



Oxtel Series Automation Protocol

**01035-15
© May 2009**

Miranda Technologies Inc.

UK

Tel: +44 (0) 1491 820000
Fax: +44 (0) 1491 820001
Email: uksales@miranda.com
Web: www.miranda.com
Software: <http://www.miranda.com/portal/downloads.php>

Copyright © 1999-2009 Miranda Technologies Inc.. All rights are reserved and, under the copyright laws, this manual may not be reproduced in any form, in whole or in part, without the prior written consent of Miranda Technologies Inc..

Printed in the UK.

Preface

Warranty

Miranda Technologies Inc. offer a 24 months free parts and labour warranty on Oxtel Series equipment from the date of the initial invoice, providing there is no evidence of negligence, abuse or failure to reasonably follow instructions given by the Company for its proper use.

During the warranty period, Miranda Technologies Inc. will replace or repair at its cost, Oxtel Series equipment returned to the factory, carriage and duty paid and insured by the sender.

Before returning any item for warranty repair, a valid returns authorisation must be granted by Miranda Technologies Inc..

All repaired goods will be delivered (as instructed) carriage and duty paid and insured by the sender.

Any warranty, over and above that offered here, is the responsibility of the local Miranda Sales Office or appointed Distributor.

Contact Information

For service, repair and warranty information and for returns authorisation contact:

oxtelsupport@miranda.com

Associated Publications

Description	Part number
Presmaster Automation Protocol	01235
Imagestore User Manual	02359
Imagestore 3 & 2U User Manual	02954
Imagestore HD-TV User Manual	02901
Imagestore 300 User Manual	03402
Imagestore Intuition+ User Manual	03457
Imagestore 300+ User Manual	03753
Imagestore 750 User Manual	M872-9900
LGK-3901/DSK-3901 User Guide	M887-9900

Contacts

For technical assistance, please contact your nearest Miranda Technical Support centre:

Americas (9:00am - 9:00pm EST)

Telephone +1-800-224-7882

Fax: +1-514-335-1614

techsupp@miranda.com

Europe, Middle East, Africa, UK (9:00am - 6:00pm GMT)

Telephone +44 (0) 1491 820222

Fax: +44 (0) 1491 820002

eurotech@miranda.com

France (9:00am - 5:00pm GMT+1)

Telephone +33 1 55 86 87 88

Fax: +33 1 55 86 00 29

eurotech@miranda.com

Asia (9:00am - 5:00pm GMT+8)

Telephone +852-2539-6987

Fax: +852-2539-0804

asiatech@miranda.com

China

Telephone: +86-10-5873-1814

asiatech@miranda.com

www.miranda.com

Table of Contents

Preface	3
Warranty -----	3
Associated Publications -----	4
Contacts -----	5
Table of Contents	7
Introduction	21
Applicability-----	21
Overview -----	22
Important Notes -----	22
Interconnections	23
Cable-----	23
Pin Assignments -----	23
RS232 -----	23
RS422 -----	24
Data Link Layer	25
Serial Data Layout-----	25
Data Transfer Rate -----	25
Coded Binary -----	25
Data -----	25
Control Codes -----	25
CRC Generation -----	26
Transport Layer	27
RS232/RS422 Point-to-Point Packet Flow-----	27
Command List -----	27

STX0 / STX1 -----	27
ACK / NAK -----	28
Command Terminator -----	28
Command Separator -----	28
Status Responses -----	28
Unsolicited Messages (Tallies) -----	29
TCP/IP-Based Automation	31
Serial Automation-----	31
Network Automation -----	31
Command List -----	32
Status Response -----	32
Unsolicited Messages -----	32
Command List	33
Video Commands-----	33
Audio Commands-----	33
Status Responses -----	33
Extended Commands and Responses -----	34
Video Commands	35
Layer Selections -----	36
Swap Preview Mode -----	36
Cascade Mode -----	36
Transition Commands -----	37
Fade to Black / Fade from Black 0	37
Fade Keyer Up / Fade Keyer Down 1	37
Cut to Black / Cut from Black 2	38
Cut Keyer Up / Cut Keyer Down 3	38
Swap Preview and Main Image using Cut 6	39
Swap Preview and Main Image using V-fade 7	39
Set Fader Angle @	39
Set Transition Duration B	40
Image Commands -----	41
Load Image from Library 8	41

Save Image to Library 9 -----	42
File Save Masked I -----	42
Enquire File Status P -----	43
Erase File Q -----	43
Load / Unload Live Video H -----	43
Grab Fill/B+KEY to the Preview Store J -----	44
Erase Store A -----	44
Enquire Loaded Image Status O -----	45
Input Commands -----	47
Force Colour Field L -----	48
Pass SDI K -----	49
Masking & Positioning Commands -----	50
Set Layer Mask 4 -----	50
Layer Mask Enable 5 -----	50
Set Image Position G -----	51
Set Input Masks I -----	52
Keyer Control Commands -----	54
Set Self Key, Separate Key, or No Key C -----	54
Set Key Linear or Full D -----	55
Set Key Normal or Invert E -----	56
Set Clip, Gain and Transparency F -----	56
Miscellaneous Commands -----	58
Enquire System Status M -----	58
Enquire Video Layer Status N -----	59
Set Preview Keyer Y1 -----	61
Set Preview Mixer Y3 -----	61
Select Channel Input UF -----	62
Tally Commands -----	63
Enable Video Tallies Y6 -----	63
Request Wake-up Packet Y7 -----	66
Wake-up Packet Y7 -----	66
Enable Audio Tallies Y8 -----	67
Image Load Tallies Y9 -----	68
Image Preload Tally YA -----	69
Enable Media Tallies YB -----	69
Heartbeat Tally YD -----	71
Loaded Image Status Tally YO -----	71

Fade-To-Black State Tally	Yf	-----	72
Text Crawl Count Tally	Yg	-----	72
Animation/Clip Cycle Tally	Yj	-----	73
Keyer Position Tally	3	-----	74
Audio Commands (old)		75	
Audio Commands Table		-----	76
Matrix Input Preselect	a	-----	77
Set Input Mode	b	-----	77
Set Input Pan	c	-----	77
Set Mix	d	-----	78
Set Mix Mode	e	-----	78
Set Output Patch	f	-----	78
Set Output Gain	g	-----	79
Set Embedded Output	h	-----	79
Parameter Ranges & Meanings		-----	80
phys_io		-----	80
io_mode		-----	81
gain_val		-----	81
duration		-----	81
mix_blk		-----	81
patch_source		-----	82
Easysound Automation Block Diagram		-----	83
Extended Commands		85	
Filename-Based Commands		87	
Load Image	R0	-----	87
Save Image	R1	-----	89
Erase Image	R2	-----	89
Enquire File Info	R3	-----	90
Query First File	R4	-----	90
Query Subsequent File	R5	-----	91
Enquire Extended File Info	R6	-----	92
Preload Image	R7	-----	93
Enquire Audio File Info	R8	-----	94
Emergency to Air	R9	-----	94

Image Load Mode	Rm -----	95
Validate Template	RA -----	96
Get Audio File Duration	RB-----	97
Animation Commands		99
Start Animation	S0 -----	99
Stop Animation	S1 -----	99
Select Animation Frame	S2 -----	100
Run Pan Effect	S3 -----	100
Restart Animation	S4-----	101
Bugclock Commands		103
Stop Timer	T0 -----	103
Start Timer	T1-----	103
Reset Timer	T2-----	104
Set Timer to Value	T3-----	104
AB Mixer Commands		105
Cut to A	U0 -----	105
Cut to B	U1 -----	105
Fade to A	U2 -----	105
Fade to B	U3 -----	106
Cut AB	U4 -----	106
X-Fade AB	U5-----	106
Set Transition Type	U6 -----	106
Wipe Softness Adjustment	U7 -----	107
V-Fade AB	U8-----	108
Set Absolute Mix	U9-----	108
Asymmetric Transition	UA-----	108
Transition Complete	UB-----	108
Fade to Specified Position	UC-----	109
Set V-Fade Colour	UD-----	109
Select Mixer	UE-----	109
Enquire Mix Mode	Ua-----	110
Set AB Cut Mode	Ub-----	111
DVE Hardware Options		113

Squeezy Commands	115
Set Squeeze Preset W0-----	115
Select Squeeze Configuration W1-----	116
Run Squeeze Move W2-----	117
Select Move Profile W3 -----	117
Select Preset Borders W4 -----	118
Enquire Preset W5 -----	118
2D/3D DVE Commands	119
Backwards Compatibility Issues with Squeezy-----	119
Select 2D/3D DVE Configuration W1 -----	120
Load and Run Sequence W2 -----	121
Run Sequence W7 -----	122
Run Sequence in Reverse W8 -----	123
Select 2D/3D DVE Preview Configuration WP -----	123
Select DVE Routing Input WR-----	124
How to use 2D/3D DVE via Automation -----	125
Easyplay Commands	127
Load Audio Clip k0 -----	127
Start Playout k1 -----	127
Stop Playout k2 -----	128
Unload Audio Clip k3-----	128
Enquire Audio Clip k4-----	128
Extended Status Commands	129
Temperature Monitoring X0 -----	129
Enquire Licence Code X1-----	129
Insert Log Message X2-----	130
Enquire Command Availability X3 -----	130
Unit Bypass X4 -----	130
Return Input Colour Field Values X5-----	131
Enquire Voltages X6 -----	132
Virtual GPI X7-----	133
GPI Input Status X8 -----	133
GPI Special X9 -----	135
Enquire Extended Intuition Layer status XA-----	135

GPI Output Status	XB-----	136
Serial Baud Rate	XC-----	137
Time Code Source	XE-----	138
Set Preview	XH-----	139
Set Reference	XI-----	140
Clock Offset	XJ-----	141
Get Serial Number	XK-----	142
System Halt	XL-----	143
Input Mode	XM-----	143
Serial Protocol	XN-----	145
Clean Feed	XP-----	146
Get Total Fill/Key	XR-----	147
Get Total Preview	XS-----	147
Monitor Feed	XT-----	148
Get Extended Serial Number	XU-----	149
Get Extended System Health	XV-----	149
Get Input Status	XX-----	151
Pass Close Caption	Xc-----	152
Set Time/Date	Xe-----	152
Store-Keypewriter Configuration	Xi-----	153
Set Standard	Xs-----	154
Set Frame Delay	Xt-----	154
Get AES Audio Signal Type	Xu-----	155
Get Embedded Audio Signal Type	Xv-----	156
Import Configuration File	Xx-----	156

Audio Commands 159

Inputs	j0 -----	160
Shuffle	j0 0 -----	161
Mute	j0 1 -----	162
Trim	j0 2 -----	162
Phase	j0 3 -----	163
Level	j0 4 -----	163
Input Source	j0 5 -----	164
Outputs	j1-----	165
Output Level	j1 0 -----	165
Output Level via Gain Type	j1 3 -----	166
Output Mute	j1 1 -----	166
Output Embed	j1 2 -----	167

Voice-Overs j2-----	167
Duck Preset j2 0-----	169
Voice-over Preset j2 1 -----	169
Voice-over Level j2 2-----	170
Rates j3 -----	170
Fade to/from Silence j30 -----	171
Fade A/B j31 -----	171
Fade Voice-over 1 Pair 1 j32-----	171
Fade Voice-over 1 Pair 2 j33-----	172
Fade Voice-over 2 Pair 1 j34-----	172
Fade Voice-over 2 Pair 2 j35-----	173
Transitions j4-----	173
Cut A/B j40 -----	173
Fade A/B j41 -----	174
Cut to/from Silence j42-----	174
Fade to/from Silence j43-----	174
Fade Voice-over 1 From/To Silence j44-----	175
Fade Voice-over 2 From/To Silence j45-----	176
Cut Voice-over 1 j46-----	176
Cut Voice-over 2 j47-----	177
Audio Follow Video j5-----	178
Audio Follow Fade to Black j50-----	178
Audio Follow A/B j51-----	178
Audio Follow Keyers j52-----	179
Voice-over Follow Easyplay j53-----	179
Audio Preview j6-----	179
Audio Enquire j7-----	181
Enquire Inputs j70 -----	181
Enquire Voice-over j71 -----	182
Enquire Outputs j72 -----	183
Enquire Rates j73-----	184
Enquire Audio Follow Video j74-----	184
Enquire Preview j75-----	185
Enquire Clean Feed j76 -----	186
Enquire Monitor j77-----	187
Enquire Output Gain j79-----	187
Audio FX Send/Return and Data Paths-----	188
FX Send Setup j8-----	188
Audio Data Path Setup j9-----	189
Enhanced AB Mixer Controls-----	190

Audio AB Position ja-----	190
Audio AB Mix Mode jb -----	190
Audio AB Asymmetric Transition jc-----	190
Audio AB Fade to Position jd-----	191
Audio A and B Fade Position je & jf-----	191
Set Audio Input Shuffles jg-----	191
Audio Clean Feed Select jh-----	192
Audio Monitor Feed Select ji-----	192
Setup Audio Delay Bank jl-----	192
Audio Metering Commands	193
Set Audio Metering Mode Y0-----	193
Audio Metering Packet Y0 -----	194
Dolby and Up-mix Commands	195
Set Up-Mix Mode q00-----	195
Set Dolby E Encoder Mode q10-----	195
Set Dolby Digital Encoder Mode q20-----	195
Set Dolby Decoder Mode q30-----	196
Metadata Commands	197
Set Pitch Shift Code o0-----	197
Set Program Description o1 -----	197
Set Program Bitstream Mode o2-----	198
Set Program Audio Coding Mode o3-----	199
Set Program Centre Down-Mix Level o4-----	199
Set Program Surround Down-Mix Level o5-----	200
Set Program Dolby Surround Mode o6-----	200
Set Program LFE Enable o7-----	201
Set Program Dialogue Normalisation o8-----	201
Set Program Production Mix Level o9-----	202
Set Program Production Room Type oA-----	202
Set Program Copyright Flag oB-----	203
Set Program Original Bitstream Format oC-----	203
Set Program Down-Mix Mode oD-----	204
Set Program Down-Mix Lt/Rt Centre Mix Level oE-----	204
Set Program Down-Mix Lt/Rt Surround Mix Level oF-----	205

Set Program Down-Mix Lo/Ro Centre Mix oG-----	205
Set Program Down-Mix Lo/Ro Surround Mix Level oH-----	206
Set Program Dolby Surround EX™ Mode ol-----	206
Set Program Dolby Headphone Mode oJ-----	207
Set Program A/D Converter Type oK-----	207
Set Program High Pass Filter Enable oL-----	208
Set Program Low Pass Filter Enable oM-----	209
Set Program LFE LPF Enable oN-----	209
Set Program Surround 90 Degree Phase-Shift oO-----	210
Set Program Surround 3dB Attenuation oP-----	210
Set Program RF Pre-emphasis oQ-----	211
Set Program Compression Profile/Word oR-----	211
Set Program Dynamic Range Value oS-----	212
Load Metadata Preset oT-----	213
Set Program Dynamic Range Type oU-----	213

Easytext Commands 215

Introduction-----	215
Summary -----	215
International Character Set Support -----	216
More Sophisticated Control of Easytext-----	217
Sending Colours via Automation-----	217
Main Commands-----	218
Update Text Field Z0-----	218
Change Box Size and Position Z1-----	219
Render Box Z3-----	219
Change Image Z4-----	220
Set Text Font and Colour Z2-----	220
Set Transparency ZA-----	221
Set Text Drop Shadow Z9-----	222
Set Text Tracking ZB-----	222
Set Text Alignment ZE-----	223
Set Text Wrapping ZF-----	223
Text Box Background Settings-----	224
Set Text Background to Clear Z8-----	224
Set Text Background to Matte Z6-----	224
Set Text Background to Gradient Z7-----	225

Setting Template background options	ZD-----	226
Text Straps	-----	227
Run Strap	Z5-----	227
Set Strap Speed	ZC-----	227
Enquire Crawl Count Loop	Zc-----	228
Text Box Update	Zd-----	229
Image Update	Ze-----	230
Enquire Textbox	Za-----	230
Enquire Template	Zb-----	231
Stop Animation	Zf-----	232
Pause/Restart Animation	Zg-----	232
Set Text Crawl Count	Zh-----	232
Datasources	-----	234
Datasource Automation Commands	-----	236
Set Global Datasource Value	m0-----	236
Delete Global Datasource	m1-----	237
Set Layer-Specific Datasource Value	m2-----	237
Delete Layer-Specific Datasource	m3-----	237
List Global Datasources	m4-----	237
Query Global Datasource	m5-----	238
List Layer-Specific Datasources	m6-----	238
Query Layer-Specific Datasource	m7-----	239
Delete All Datasources	m8-----	239
Internal Datasources	-----	240
TIME	-----	240
DATE	-----	241
FADER0 through FADER7	-----	241
HOSTNAME	-----	241
IP_ADDR	-----	242
SERIAL	-----	242
SYSNAME	-----	242
MPEG Commands		243
MPEG Control	v0-----	243
MPEG Load	v1-----	243
EAS Commands		245

EAS On n1 -----	245
EAS Off n2 -----	245
EAS Set Text n3 -----	245
EAS Set Priority n4 -----	246
Passive Mode Commands	247
Set Passive Mode On/Off Ya -----	247
Set Passive Mode Transmission Interval Yb-----	247
Layer Rate Yc-----	248
Strap Loop Count Yd-----	248
Enquire Passive Mode Ye -----	248
Scheduled Commands	249
Add Scheduled Command i0 -----	249
Timecode -----	250
Delete Scheduled Command i1 -----	250
Delete All Scheduled Command i2 -----	250
Time Of/Till Next Event i3-----	251
IntuitionXG Commands	253
Cue V0-----	253
Take V1 -----	253
Clear V2 -----	254
Keyer Cut/Fade Up V3 -----	254
Keyer Cut/Fade Down V4-----	254
Set Property V5 -----	255
Enable IntuitionXG Tallies V8-----	256
Miscellaneous Commands	257
Image Count Ra-----	257
Pixel Count Rb -----	257
Appendix A Command Validity	259
Appendix B Layer Numbers	269
Imagestore IS1 & Imagestore IS2 -----	269

Imagestore Upstream-----	269
Imagestore 300[+] & Imagestore HD-TV-----	269
Intuition[+]-	269
Imagestore 750-----	269
LGK-3901-----	270
DSK-3901-----	270
Appendix C CRC Generation	271
Appendix D Automation Examples	273
Keyer Animation Cut/Fade-----	273
AB Mixing -----	273
DVE Moves-----	274

Introduction

Applicability

This user guide describes the automation control protocol used by Miranda Technologies Inc. Oxtel range of broadcast products. The Oxtel series products that can be controlled using this protocol include:

- Imagestore 2 (v2.25)
- Imagestore 2U (v2.25)
- Imagestore 3 (v2.25)
- Imagestore HD-TV (v1.13.8)
- Imagestore 300 (v1.13.8)
- Imagestore 300+ (v1.13.8)
- Intuition SD (v1.13.8)
- Intuition HD (v1.13.8)
- Intuition SD+ (v1.13.8)
- Intuition HD+ (v1.13.8)
- Imagestore 750 (v2.0.1)
- Intuition XG
- LGK-3901 (v2.1)
- DSK-3901 (v2.1)

This document describes all automation commands for every version of each Oxtel unit and is accurate for the latest production software releases at the time of this publication (software versions shown in brackets). Individual differences between units are highlighted in the appropriate section and the Command Validity Table on page 259.

The Presmaster Master Control Switcher uses a different protocol – ask for Presmaster Automation Protocol document (part no. 01235) for details.

Overview

The Oxtel unit takes the role of the controlled device, while the automation system is the controller. Automation commands are sent to the Oxtel unit from the controller, and the Oxtel unit actions these commands returning status information as requested.

With RS232 or RS422 communications, the link between a controlled device and its controller is point-to-point (a single physical connection between the two).

With later Imagestore products (IS2/2U/3 v2.08 and all products listed below Imagestore 3 on the list on page 21) automation control via TCP/IP is also available.

Important Notes

The RS232 pin-out follows the conventionally accepted standard. For Imagestore 2-based products, and earlier the RS422 pin-outs are based on industrial automation standards, and do not conform to Sony “9-Pin” or ES-Bus designations. Newer products including Imagestore HD-TV, Imagestore 300, Imagestore Intuition, Imagestore 750, LGK-3901 and DSK-3901 comply with the 9-pin standard. Please refer to individual product User Manuals for details.

Interconnections

Automation control of Oxtel products uses RS232 or RS422 asynchronous serial communications.

All Imagestore units have two or more connectors with one being RS232, and another being RS422 (point-to-point). The connectors on later Imagestore products, for example Imagestore 750, LGK-3901 and DSK-3901 are configurable to different serial types including RS232 and RS422. Set up of serial ports (i.e. protocol, baud rate, and for later products, serial type) is covered in individual User Manuals for each product.

Cable

All interconnection cables attached to Oxtel units should be constructed following best practice for the purposes of electromagnetic compatibility (EMC).

Cables should always feature a screening braid fabrication, and always be terminated with metal shrouded connectors. The fully screened cable must have a metal/foil shroud completely surrounding the internal wiring within the terminating connectors. This shroud must be connected directly to the cable's screening braid. This connection must be as close to 360 degrees around the cable as possible. Connections using thin wires or drain wires should not be used, since they provide inadequate screening.

To conform with broadcast standard earthing/grounding arrangements, the cable screen should only be connected to ground at one end via the connector shroud.

Pin Assignments

RS232

The RS232 connector is a 9-way male D-type, which follows the same pin assignments as the standard IBM PC/AT serial port, as shown in the following table:

Pin	Function
2	RX
3	TX
5	GND

All other pins are not used and must be unconnected.

RS422

The RS422 pin-out is a 9-way male D-type with the following pin-out:

Pin	Function
1	TX -
2	TX +
3	RX +
4	RX -
5	GND

All other pins are not used and must be unconnected.

Note: For Imagestore 2-based products and earlier the RS422 pin-out does NOT conform to the broadcast industry standard commonly known as “Sony 9-pin”. Newer products including Imagestore HD-TV, Imagestore 300, Imagestore Intuition Imagestore 750, LGK-3901 and DSK-3901 comply with the 9-pin standard.

The next table shows a typical point-to-point RS422 connection:

Controller	Controlled Oxtel Series Unit
TX -	RX -
TX +	RX +
RX +	TX +
RX -	TX -
Ground	Ground

Data Link Layer

Serial Data Layout

1 start bit, 8 data bits, 1 stop bit. No parity bit.

Data Transfer Rate

Baud rates of 9600, 19200, 38400, 57600, 115200 can be selected for any RS232 or RS422 serial port.

For some much older units, the data rate for communication is 19200 baud irrespective of interface standard. Please refer to individual User Manuals for details.

Coded Binary

Data

Apart from the CRC, ACK, NAK and STX characters, all Oxtel Protocol characters are in the range of printable ASCII. ($\geq 0x20$).

Control Codes

The control codes used are defined in the following table.

Code	Value
STX0	0x2
STX1	0x3
ACK0	0x4
ACK1	0x5
STX2	0x6
NAK	0x7

Note: The names for these codes do NOT correspond to ASCII code naming conventions.

CRC Generation

The CRC (Cyclic Redundancy Check) bytes provide a means of verifying the transmission and receipt of incoming packets.

A 16-bit CRC value should be generated (by Automation) using the contents of the corresponding transmission packet. The resultant value is appended to the data packet and transmitted to the Oxtel unit as two 8-bit bytes. Using the same CRC algorithm at the receiving end, the Oxtel unit is able to verify the integrity of the data transmitted.

The same CRC technique is used on the reverse channel by the Oxtel unit when it returns transmission packets back to Automation.

The CRC is constructed from all characters that follow the STX. A software coding example in Appendix C (page 271) shows how to generate and use CRCs.

Transport Layer

RS232/RS422 Point-to-Point Packet Flow

Command List

For transmission over point-to-point links, the command list is encapsulated in packets as described below:

Field	Contents
Byte 0	STX0/1
Byte 1..n-1	Command List
Byte n	Command Terminator ‘:’
Byte n+1	CRC low
Byte n+2	CRC high

STX0 / STX1

Oxtel units expect the STX values in successive packets that are transmitted by Automation to alternate between STX0 and STX1.

On receipt of a good packet the Oxtel unit will transmit a single character response of either ACK0 or ACK1 – (where the ACK matches the STX that was originally sent).

Automation should wait for an ACK before sending the next packet. However if no ACK is received, then Automation must resend the same message with the same STX value. This procedure means that lost ACKs will not cause message duplication.

If message transmission fails (i.e. all retries either receive a NAK response or they time-out), then the next attempt to send a message should use the alternate STX value. This is important in older products and software, as a system reset could cause the Oxtel unit to be waiting for the wrong STX. However in later software either STX is accepted for:

- The first packet at start-up
- The packet following one with a CRC error or incorrect STX

ACK / NAK

If the STX is wrong then neither ACK nor NAK is returned and Automation will have to deal with a time-out.

If the STX was correct but the CRC wrong, then a NAK is sent back to Automation by the Oxtel unit

If both the STX and CRC are correct for a packet then an ACK0 or ACK1 character (corresponding to the STX) is sent back to Automation. ACKs will be returned even if the command list contains invalid commands or parameters, or errors occur during the processing of the commands.

Command Terminator

Since the command terminator character “:” (colon) marks the end of a command list, this character must not occur within the command list. When a command list includes a string parameter that needs a colon character (for example, the m003 command), it can be escaped as “\3A”.

Command Separator

Multiple automation commands may be included within a single command list by concatenating them with the command separator character “;” (semicolon). When a command list includes a string parameter that needs a semi-colon character, it can be escaped as “\3B”.

Status Responses

Status requests from the controller cause the following message to be returned:

Field	Contents
Byte 0	STX0/1
Byte 1..n-1	Status Response
Byte n	Status Terminator “:”
Byte n+1	CRC low
Byte n+2	CRC high

Status messages must not be acknowledged by Automation, and cannot be retransmitted. The STX value sent matches the STX value of the original status request. If Automation fails to receive a status message it must request the status again.

As the STX of the response matches the STX of the command, sending a command list containing multiple status requests will cause multiple responses with identical STX bytes.

Unsolicited Messages (Tallies)¹

Unlike status replies, which are only returned in response to particular commands, unsolicited messages can be sent by the Imagestore at any point based on changes in state. Unsolicited messages are distinguished by the leading STX2 byte.

Field	Contents
Byte 0	STX2
Byte 1..n-1	Status Response
Byte n	Status Terminator ‘:’
Byte n+1	CRC low
Byte n+2	CRC high

Various types of unsolicited messages are generated by Imagestore, including audio metering data, video and audio tallies, and other status information. The transmission of metering data and tallies must be explicitly enabled by the automation system.

¹ Supported in Imagestore 2/2U/3 version 2.08 and later, plus all newer products

TCP/IP-Based Automation

Serial Automation

Most Oxtel products² have the ability to run automation protocols over a standard TCP/IP socket interface, provided the unit is equipped with Ethernet interface. The packet layout and protocols are exactly as described above for RS232/RS422 point-to-point packet flow, including STX and CRC checksum bytes.

Connections are accepted on port 5006. Multiple connections can be established to the Imagestore, allowing distributed or backup control without the need for additional cabling.

When considering using TCP/IP automation, please note the following:

- Any Ethernet-based protocol cannot guarantee frame-accurate reception and there may also be non-deterministic latencies due to network traffic.
- Each TCP/IP connection will have its own parameters for information such as tally responses and audio metering. By default these are turned off on new connections, and need to be enabled (if required) whenever a new connection is established.
- Imagestore may limit the number of simultaneous sessions.

Network Automation

The Linux-based Imagestore products (Imagestore HD-TV, Imagestore 300, Imagestore Intuition, Imagestore 750, LGK-3901, DSK-3901) will additionally accept Network Automation commands on port 5007.

Network Automation is a simplified version of Serial Automation which takes advantage of TCP's end-to-end reliability. It differs as follows:

- It does not use an STX byte.
- It does not use CRC bytes.
- No ACK/NAK byte is returned.

Otherwise the command list and status response format is exactly the same. In particular the command terminator ':' (colon) is still required to mark the boundaries in the byte stream.

² Supported in Imagestore 2/2U/3 version 2.08 and later, plus all newer products

Command List

Field	Contents
Byte 1..n-1	Command List
Byte n	Command Terminator ‘:’

Status Response

Field	Contents
Byte 1..n-1	Status Response
Byte n	Status Terminator ‘:’

Unsolicited Messages

Field	Contents
Byte 1..n-1	Status Response
Byte n	Status Terminator ‘:’

Command List

Video Commands

Video layer commands are represented by a command list. Although parameter formats do vary, the video layer commands are often formatted with parameters separated by spaces.

cmd	param_1	space	param_2	space	param_n
-----	---------	-------	---------	-------	---------

The terms used in the above command list are expanded below:

Term	Description
cmd	The command code, a single ASCII character
param_1	
param_2	
param_n	The parameters for the particular command code
Space	A space character (ASCII code 32 or 0x20).

Please see page 35 for details on individual video commands.

Audio Commands

Audio layer commands use the same scheme as the video commands above, except that parameters are not separated by spaces. This is because all parameters are of fixed length.

Example 1:

cmd	param_1	param_2
-----	---------	---------

Example 2:

cmd	param_1	param_2	param_3
-----	---------	---------	---------

Status Responses

Commands that require a status response will cause status response packets to be generated in the following format:

cmd	param_1	param_2	param_3	param_n
-----	---------	---------	---------	---------

In the above format, **cmd** is the echoed status enquiry command. The actual number and types of parameters returned is dependent on the command type issued, so please check individual commands for formatting.

Extended Commands and Responses

These messages have two bytes defining the command, allowing a much wider range of commands. The first byte typically specifies a command group, while the second identifies a sub-command. Parameter formatting depends on the command, and so details of each command should be checked.

Note: Command characters are always case-sensitive.

Video Commands

ID	Description	Type
0	Fade to Black / Fade from Black	Transition
1	Fade Keyer Up / Fade Keyer Down	Transition
2	Cut to Black / Cut from Black	Transition
3	Cut Keyer Up / Cut Keyer Down	Transition
4	Set Mask	Masking & Positioning
5	Mask Enable / Disable	Masking & Positioning
6	Swap Preview and Main Image using Cut	Transition
7	Swap Preview and Main Image using V-fade	Transition
8	Load Image from Library	Image
9	Save Image to Library	Image
<	Set Transition Type	Transition
@	Set Fader Angle	Transition
A	Erase Store	Image
B	Set Transition Duration	Transition
C	Set Self Key, Separate Key, or No Key	Keyer Control
D	Set Key Linear or Full (key affects fill)	Keyer Control
E	Set Key Normal or Invert	Keyer Control
F	Set Clip, Gain and Transparency	Keyer Control
G	Set Image Position	Masking & Positioning
H	Load / Unload Live Video	Image
I	File Save Masked	Image
J	Grab Fill/B+KEY to the Preview Store	Image
K	Pass SDI	Input Control
L	Force Colour Field	Input Control
M	Enquire System Status	Miscellaneous
N	Enquire Loaded Image Status	Miscellaneous
O	Erase File	Image
P	Enquire File Status	Image
Q	Erase File	Image
I	Set Input Masks	Masking & Positioning
Y1	Set Preview Keyer	Miscellaneous
Y3	Set Preview Mixer	Miscellaneous

Layer Selections

The layer parameter passed to many commands can have different meanings depending on the unit being controlled, and its current operating mode.

Swap Preview Mode

For units in swap/preview mode, where one image is keyed onto the preview output and a second image keyed over the program output, layer selection works like this:

- Layer 0 = preview layer
- Layer 1 = program layer

This allows independent control of both layer outputs. Automation systems can effectively configure the units as two-output devices.

Cascade Mode

In cascade mode (default and recommended) both keyers are used to insert independent images onto the program output. Preview ability is more limited in this mode, but more keying layers can be viewed. In cascade mode, layer 0 is DSK1 (often called the *midground* layer), and layer 1 is DSK2 (often called the *foreground* layer).

- Layer 0 = DSK1 (midground layer)
- Layer 1 = DSK2 (foreground layer)

Please note that midground/foreground terminology is not applicable to Imagestore 750, LGK-3901, or Imagestore Intuition+ since these products have more than two keying layers – (4, 5 and 16 layers respectively).

Note: Further details on the meaning of layer numbers for different products can be found on page 269.

Transition Commands

Fade to Black / Fade from Black 0

These commands cause the total output from the selected keyer layer to be faded to or from black. Repeated commands of the same sort have no effect. If the command is not finished when a reverse instruction is received, the fade will reverse direction and continue at the same rate. The transition duration of the fade to/from black can be set using the B command or by using the optional third parameter for Imagestore 300[+], Imagestore HD-TV, Imagestore 750, LGK-3901 and DSK-3901. The value provided sets the number of fields to complete the action. This value may range from 0 to 999 decimal, but is encoded in the command as hexadecimal.

CMD	Param_1	Param_2	Param_3 (optional)
0	Layer No.	0=To black, 1=From black 2=Fade to/from black	Rate in fields (1-999)

Example:

```
void FadeToBlack(int Layer, bool Black)
{
    remote_send("0%x %d", Layer, Black ? 0:1);
}
```

Fade Keyer Up / Fade Keyer Down 1

These commands cause the keyer layer to be faded up or down. When the keyer is faded down, the background is passed unchanged. Repeated commands of the same sort have no effect. If the command is not finished when a reverse instruction is received, the fade will reverse direction and continue at the same rate. The transition duration of the fade keyer up/down can be set using the B command or by using the optional third parameter for Imagestore 300[+], Imagestore HD-TV, Imagestore 750, LGK-3901 and DSK-3901. The value provided sets the number of fields to complete the action. This value may range from 0 to 999 decimal, but is encoded in the command as hexadecimal.

CMD	Param_1	Param_2	Param_3 (optional)
1	Layer No.	0=Down, 1=Up 2=Fade up/down	Rate in fields (1-999)

Example:

```
void FadeKeyer(int Layer, bool Up)
{
    remote_send("1%x %d", Layer, Up ? 1:0);
}
```

Cut to Black / Cut from Black 2

These commands are identical to an instant fade to / from black.

CMD	Param_1	Param_2
2	Layer No.	0=To black, 1=From black 2=Cut to/from black

Example:

```
void CutToBlack(int Layer, bool Black)
{
    remote_send("2%x %d", Layer, Black ? 0:1);
}
```

Cut Keyer Up / Cut Keyer Down 3

These commands are identical to an instant fade in the designated direction.

CMD	Param_1	Param_2
3	Layer No.	0=Down, 1=Up 2=Cut up/down

Example:

```
void CutKeyer(int Layer, bool Up)
{
    remote_send("3%x %d", Layer, Up ? 1:0);
}
```

Swap Preview and Main Image using Cut 6

Note: This command is largely deprecated.

Also known as “Cut Swap”

This command is only relevant to units used in swap-preview mode. The output is cut down, the preview store is then swapped with the main store and the keyer output is cut back up to maximum. If the main store was faded down prior to this action, then this part of the operation is ignored. The switching action takes place within the vertical blanking interval and the background is not visible between the cut down and the cut up.

CMD
6

Swap Preview and Main Image using V-fade 7

Note: This command is largely deprecated.

Also known as “Fade Swap”

This is only relevant to units running in swap-preview mode. The output is faded down, the preview store is then swapped with the main store and the keyer output is faded back up. If the main store was faded down prior to this action, then this part of the operation is ignored.

CMD
7

Set Fader Angle @

This command causes the keyer layer fader to be set to an absolute angle. Type is a single digit defining which transition is to be affected, the selection being between Fade and Fade-To-Black (FTB). Angle is a value between 0 the minimum angle, and 512 (or 0x200) the maximum angle

CMD	Param_1	Param_2	Param_3
@	Layer No.	Type 0=FTB, 1=Fade	Angle

Example:

```
void SetFaderAngle(int Layer, int Type, int Angle)
{
    remote_send( "@%x %d %x", Layer, Type, Angle);
}
```

Set Transition Duration B

This command sets the duration for the selected transaction on the selected layer. In swap-preview mode, the value applies to the main layer, whatever layer is actually selected. The value provided sets the number of fields to complete the action. This value may range from 0 to 999 decimal, but is encoded in the command as hexadecimal. Type is a single digit defining which transition is to be controlled, the selection being between Fade and Fade-To-Black (FTB).

CMD	Param_1	Param_2	Param_3
B	Layer No.	Type 0=FTB, 1=Fade	Rate in fields

Example 1: Set transition rate.

```
void SetTransRate(int Layer, int Fields)
{
    remote_send( "B%x 1 %x", Layer, Fields);
}
```

Example 2: Set Fade-To-Black transition.

```
void SetFTBTransRate(int Layer, int Fields)
{
    remote_send( "B%x 0 %x", Layer, Fields);
}
```

Image Commands

Older Oxtel Imagestores use numbered images. Newer IS2-based units, Imagestore HD-TV, Imagestore 300, Imagestore 750, LGK-3901 and DSK-3901 can use numbered images and can also use named images, with file extensions “.OXT” for stills and “.OXA” for clocks and animations.

Imagestore Intuition uses named images with the file extension “.TEM” for templates.

Many of the commands listed in this section only work for numbered images.

When sent to IS2-based units, Imagestore HD-TV, Imagestore 300, Imagestore 750, LGK-3901 and DSK-3901 a numbered file name is constructed in the form “Vxxx.OXT” or “Vxxx.OXA”, where the xxx is the specified number in decimal. When sent to Imagestore Intuition, a numbered file name is constructed in the form “Vxxx.TEM”, where the xxx is the specified number in decimal. If the specified number in decimal has 4 digits, then the file name is constructed in the form “Vxxxx.OXT”, “Vxxxx.OXA” or “Vxxxx.TEM”.

To load and save named files (the preferred option), use the “R” family of filename-based extended commands.

Load Image from Library 8

Note: This has been superseded by the R0 command.

This reads an image into the selected store. Images are numbered from 0 to n, where n is dependent on the storage option purchased. The image number is in hexadecimal.

CMD	Param_1	Param_2
8	Layer No.	Image No. (hex)

Example: Load image V123.OXT

```
void LoadImage123(int Layer)
{
    remote_send("8%x %x", Layer, 123);
}
```

Save Image to Library 9

Note: This has been superseded by the R1 command.

This saves an image in the selected store to the internal library. Images are numbered from 1 to n, where n is dependent on the storage option purchased. The image number is in hexadecimal.

CMD	Param_1	Param_2
9	Layer No.	Image No. (hex)

Example: Save image as V123.OXT

```
void SaveImage123(int Layer)
{
    remote_send( "9%x %x", Layer, 123 );
}
```

Note: This save command only works if the image number used matches the loaded image number (loaded via the 8 command). You cannot save the current loaded image to a different file number.

File Save Masked I

Note: This command is largely deprecated.

This command saves an image in the selected store to the internal library, saving only that portion of the image that is visible through the applied mask. Images are numbered from 1 to n, where n is dependent on the storage option purchased. In IS2, the range is not limited, only the total number of images saved. The image number is sent as a hexadecimal value.

CMD	Param_1	Param_2
I	Layer No.	Image No. (hex)

Enquire File Status P

Note: This has been superseded by the R3 command.

This command returns status on the specified file slot. In particular it determines if the file slot is available in the internal image library. Images are numbered from 0 to n, where n is dependent on the storage option purchased. The image number is sent as a hexadecimal value.

CMD	Param_1
P	Image No. (hex)

The information returned is:

Format	Field	Description
%c	Cmd	
%1x	File exists	1 = file exists, 0 = file not present
%03x	File slot number	

Erase File Q

Note: This has been superseded by the R2 command.

This command erases the image file slot selected. Images are numbered from 0 to n, where n is dependent on the storage option purchased. The image number is sent as a hexadecimal value.

CMD	Param_1
Q	Image No. (hex)

Load / Unload Live Video H

This command loads (or unloads) ‘live’ video from the fill and key SDI inputs into the keyer. ‘Unloading’ live is usually unnecessary, as loading any other image will implicitly unload any live image.

CMD	Param_1	Param_2
H	Layer No.	0 = Unload Live 1 = Load Live (F1/K1) 2 = Load Live (F2/K2) 3 = Load Live (F3/K3)

Please note that setting the second parameter as 2 or 3 requires that the Imagestore hardware supports additional external Fill and Key pairs.

Example:

```
void LoadLive(int Layer, int Input)
{
    remote_send( "H%x %x", Layer, Input );
}
```

Grab Fill/B+KEY to the Preview Store J

Note: This command is largely deprecated.

This command causes a single frame of the fill and key signals present on inputs Fill/B and KEY to be stored into the selected store. The signal on Fill/B is taken as the YUV 4:2:2 signal. The Y component of the signal on KEY is taken as the key signal.

In swap/preview mode this operation is always possible. In cascade mode the midground must be cut down to 0 for this operation to work.

CMD	param_1
J	Layer No.

Erase Store A

This command empties a store and its associated key store. Both picture and key stores are filled with colour black.

On more recent Imagestore units, an *empty* image is selected. The store remains physically unchanged.

CMD	Param_1
A	Layer No.

Example:

```
void EraseStore(int Layer)
{
    remote_send( "A%x", Layer );
}
```

Enquire Loaded Image Status O

This command returns a status block about the image loaded onto the selected layer.

CMD	Param_1
O	Layer No.

Example:

```
void EnquireLoadedImageStatus(int Layer)
{
    remote_send( "O%x", Layer );
}
```

The information returned is:

Format	Field	Description
%c	Cmd	'O'
%03x	Clip	
%03x	Gain	
%03x	Transparency	
%03x	Horizontal position	
%03x	Vertical position	
%03x	Horizontal size	
%03x	Vertical size	
%03x	Mask top	
%03x	Mask bottom	
%03x	Mask left	
%03x	Mask right	
%03x	File slot number	Set when loaded
%c	Key type	0x40 OR'd with... 0x01 for self key 0x02 for linear key 0x04 for invert 0x8 for disk busy (<i>no longer supported</i>) 0x10 for key source = none.
%1x	Masked	
%1x	(reserved)	Always Zero
%1x	Easyplay playing	1 = playing, 0 = not playing
%1x	Layer	Layer number

Note: Unsolicited O command responses on state change can be requested via the YO tally (see page 71).

Input Commands

The following Input Port numbers (corresponding to physical SDI inputs) are used for the Input Control commands.

Port Name										Imagenstore Upstream
										Imagenstore IS2
										Imagenstore IS1
										Imagenstore 300
										Imagenstore 300+
										Imagenstore HD-TV
										Imagenstore 750
A Input ³	0	X	X	X	X	X	X	X	X	
Fill-1	1	X	X	X	X	X	X	X	X	X
Key-1	2	X	X	X	X	X	X	X	X	X
B Input	3			X	X	X	X	X	Xa	
Fill-2	4	X	X	X	X	X	X			
Key-2	5	X	X	X	X	X	X			
Fill-3 (C)	6			X						
Key-3 (D)	7			X						

NOTES:

a : IS2 with A/B Mix option (MIX-2000) only.

In addition to commands listed in this section, Input Port numbers are also used by the 'I' and 'XM' commands (page 52 and page 143).

³ Also known as "Background" on some Imagestore products and Program In on LGK-3901/DSK-3901

Force Colour Field **L**

This command causes the selected input channel to produce a colour field. The R, G & B values are given as parameters. The values are percentages of saturated colour, for example 0x32, 0x32, 0x32, is 50% red, green and blue. See table at the head of this page for input port numbers.

CMD	Param_1	Param_2	Param_3	Param_4
L	Input Port No.	Red 0-x64	Green 0-x64	Blue 0-x64

See table on page 47 for input port numbers.

Example:

```
void ForceColourField(int Input,int R,int G,int B)
{
    remote_send("L%x %x %x %x", Input, R, G, B);
}
```

Imagestore 750, LGK-3901 and DSK-3901 do not have dedicated colour fields for each input. Instead it has three “shared” colour fields, each of which can be assigned to any input via the XM command (see page 143). The first parameter of the L command therefore refers to a colour field number.

- 0: Colour Field 1
- 1: Colour Field 2
- 2: Colour Field 3

CMD	Param_1	Param_2	Param_3	Param_4
L	Colour Field No.	Red 0-x64	Green 0-x64	Blue 0-x64

Pass SDI K

This command allows the selected input channel to pass the SDI (if present). If no SDI is present then the colour field will be automatically substituted.

CMD	Param_1
K	Input Port No.

See table on page 47 for input port numbers.

Example:

```
void PassSDI(int Input)
{
    remote_send( "K%x" , Input );
}
```

Note: For Imagestore 750, LGK-3901 and DKS-3901 black output will be shown when no SDI signal is present. A colour field (or test pattern) is only shown when it is explicitly routed to the input via the XM command.

Masking & Positioning Commands

Set Layer Mask 4

Loads the mask registers with the left, right, top, and bottom values defining the size of the mask. In cascade mode, the mask registers of both layers may be loaded, but in swap / preview mode, only the preview layer mask registers may be loaded.

The left and right values must be from zero to the width of the image loaded onto the layer specified (measured in pixels). Similarly the top and bottom values must be from zero to the height of the image.

CMD	Param_1	Param_2	Param_3	Param_4	Param_5	Param_6 (optional)
4	Layer No.	Left	Right	Top	Bottom	Speed

Example:

```
void SetMask(int Layer, int Left, int Right,
             int Top, int Bottom)
{
    remote_send("4%x %x %x %x %x", Layer, Left,
                Right, Top, Bottom);
}
```

Speed parameter is optional and is measured in pixels/lines per field.

Note: Speed is not implemented in Imagestore 300, Imagestore HD-TV, Imagestore 750, LGK-3901 or DSK-3901

Layer Mask Enable 5

In cascade mode, the mask registers of both layers may be enabled/disabled, but in swap/preview mode, only the preview layer mask registers may be enabled/disabled.

CMD	Param_1	Param_2
5	Layer No.	0=Disable , 1 =Enable

Example:

```
void MaskEnable(int Layer, bool Enable)
{
    remote_send("5%x %d", Layer, Enable);
}
```

Set Image Position **G**

This command sets the position of the image relative to the viewing screen. The origin is the upper left hand corner with positive increments moving the picture downwards and to the right. The units of the 'horizontal' and 'vertical' values are in pixels and lines respectively.

CMD	Param_1	Param_2	Param_3	Param_4 (optional)
G	Layer No.	Horizontal	Vertical	Speed

Example:

```
void SetImagePosition(int Layer, int Hor, int Ver)
{
    remote_send("G%x %x %x 0", Layer, Hor, Ver);
}
```

The ranges for different standards are:

PAL (625):	±720 horizontally, ±576 vertically
NTSC (525):	±720 horizontally, ±486 vertically
720p:	±1280 horizontally, ±720 vertically
1080i:	±1920 horizontally, ±1080 vertically

Speed is an optional parameter and is measured in pixels/lines per field.

Note: Speed is not implemented in Imagestore 300, Imagestore HD-TV, Imagestore 750, LGK-3901 or DSK-3901.

Pictures may be placed off screen in all four directions.

The status of the image position on each layer can be enquired using:

CMD	Param_1
G	Layer No.

Example:

```
void EnquireImagePosition(int Layer)
{
    remote_send( "G%x", Layer );
}
```

This will return:

CMD	Param_1	Param_2	Param_3
G	Layer No.	Horizontal	Vertical

Set Input Masks I

This command sets the input mask for each SDI input of an Imagestore 750. All of the mask parameters are measured in pixels so their range is dependent on the standard.

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
I	%x: Input	%x: Left	%x: Right	%x: Top	%x: Bottom

Example:

```
void SetInputMask(int Input, int Left, int Right,
                  int Top, int Bottom)
{
    remote_send( "I%x %x %x %x %x", Input, Left,
                 Right, Top, Bottom );
}
```

See table on page 47 for input port numbers.

Setting the input mask will directly affect the SDI source, so anything using that source will be affected. Note that if an input is not currently using its SDI

source (because it has been reset to a colour field or test pattern), the image will not be masked - only the SDI source is masked.

To enquire the input masks:

CMD	Param_1
I	Input

Example:

```
void EnquireInputMask(int Input)
{
    remote_send("1%x", Input);
}
```

The values returned will be of the following format:

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
I	%x: Input	%x: Left	%x: Right	%x: Top	%x: Bottom

Keyer Control Commands

Set Self Key, Separate Key, or No Key C

Oxtel series products implement keying where the Key is contained in the luminance content of the SDI signal. The Key source is selected from either the Fill input (known as Self Key) or a separate Key signal.

Self key sets the key input associated to a particular keyer layer to be derived from the luminance content (Y value) of the “Fill” signal itself.

Separate key sets the key input associated with a particular keyer layer to be derived from the luminance content (Y value) of the separate “Key” signal.

No key causes the key input associated with a particular keyer layer to be ignored. The entire fill becomes visible without transparency.

Note: In swap-preview mode this value is associated with the preview store. The value will remain with the image as it is stored or transferred to the program channel.

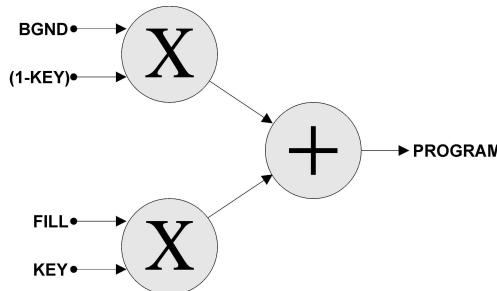
CMD	Param_1	Param_2
C	Layer No.	Type 0 = Separate Key, 1 = Self-Key, 2 = No Key

Example:

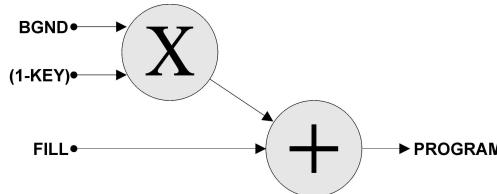
```
void SetKeyType(int Layer, int Mode)
{
    remote_send( "C%x %x", Input, Mode );
}
```

Set Key Linear or Full D

Full key refers to the standard keying algorithm where the Fill signal needs to be multiplied by the Key signal prior to adding it to the Background signal. The equation for this is: Output = Key x Fill + (1-Key) x Bgnd



Linear key is more typically called *Additive Keying*. This requires the foreground image to be ‘pre-composited’ onto a black matte. The equation for this is: Output = Fill + (1-Key) x Bgnd



CMD	Param_1	Param_2
D	Layer No.	0=Full, 1=Linear

Example:

```

void SetKeyMode(int Layer, int Mode)
{
    remote_send( "D%x %x", Input, Mode );
}
  
```

Set Key Normal or Invert E

Normal key implies that increasing key values will introduce more of the fill signal.

Key invert implies that increasing key values introduce less fill. The Key output is replaced with (1-Key), thereby inverting the action of the Key.

CMD	Param_1	Param_2
E	Layer No.	0=Normal, 1=Invert

Example:

```
void SetKeySense(int Layer, int Sense)
{
    remote_send( "E%x %x", Input, Sense );
}
```

Set Clip, Gain and Transparency F

This command sets the clip, gain and transparency values for a particular keyer layer. The parameter values are hexadecimal representations of unsigned 10-bit luminance values.

Clip is the lowest value of key signal which will cause the fill signal to contribute to the output. In a keying process this will result in a total absence of fill video in areas where the incoming key signal is less than the clip level.

Gain sets the level of key signal above which 100% of the fill signal is used. In a keying process this will result in the fill signal being less transparent than it otherwise would be in areas with a grey key signal.

Transparency controls the luminance content (grey scale value) of the key signal and therefore the amount of background video that will 'seep' through the keyed image.

Note: In swap-preview mode this value is associated with the preview store. The value will remain with the image as it is stored or transferred to the program channel.

CMD	Param_1	Param_2	Param_3	Param_4
F	Layer No.	Clip (0-1023)	Gain (0-1023)	Transparency (0-512)

Example:

```
void SetCGT(int Layer, int C, int G, int T)
{
    remote_send( "F%x %x %x %x", Layer, C, G, T);
}
```

Note: For optimum results the Clip, Gain and Transparency values should be set to 64, 940 and 512 respectively. The Clip cannot be set to greater than the Gain.
For Imagestore 300, Imagestore HD-TV and Imagestore 750, LGK-3901, DSK-3901 Clip and Gain values set between 0-63 or 941-1023 are clamped to 64 and 940 respectively.

The status of the clip, gain and transparency on each layer can be enquired using:

CMD	Param_1
F	Layer No.

This will return:

CMD	Param_1	Param_2	Param_3	Param_4
F	Layer No.	Clip (0-1023)	Gain (0-1023)	Transparency (0-512)

Miscellaneous Commands

Enquire System Status M

This command returns information about the system.

CMD
M

Example:

```
void GetSystemStatus(void)
{
    remote_send(CmdBuffer, "M");
}
```

The information returned is:

Format	Field	Description
%c	Cmd	'M'
%1d	System mode	Imagestore 2 based products: Mixer/Keyer: 0 = Keyer, 1 = Mixer Imagestore 300[+], Imagestore HD-TV, Imagestore 750, LGK-3901, DSK-3901: Still Store: 0 = Cascade, 1 = Swap Preview
%03x	Version High	
%03x	Version Low	
%1d	Video Standard	0 = PAL, 1 = NTSC, 2 = 1080i@59.94Hz, 3 = 1080i@50Hz, 4 = 720p@59.94Hz, 5 = 720p@50Hz
%03x	Preview Source	(Intuition: always 0)
%03x	Fade rate DSK1	
%03x	Fade rate DSK2	
%03x	FTB rate DSK1	
%03x	FTB rate DSK2	
%1x	System not accessed	

Enquire Video Layer Status N

This command returns a status block about the video layers.

CMD
N

Example:

```
void EnquireVideoLayerStatus( )
{
    remote_send( "N" );
}
```

The information returned for Imagestores 2, HD-TV and 300 is:

Format	Field
%c	Cmd
%03x	DSK1 Fader Angle (0x0 to 0x200)
%03x	DSK1 Fade-to-Black Angle (0x0 to 0x200)
%03x	DSK2 Fader angle (0x0 to 0x200)
%03x	DSK2 Fade-to-Black Angle (0x0 to 0x200)
%02x	AB Mix Transition Type: 0x01=V-Fade, 0x03=X-Fade, 0x05=Cut, 0x06=WipeLR, 0x07=WipeTB, 0x10=WipeRL, 0x11=WipeBT

The information returned for Imagestore 750 is:

Format	Field
%c	Cmd
%03x	DSK1 Fader Angle (0x0 to 0x200)
%03x	DSK1 Fade-to-Black Angle (0x0 to 0x200)
%03x	DSK2 Fader angle (0x0 to 0x200)
%03x	DSK2 Fade-to-Black Angle (0x0 to 0x200)
%02x	AB Mix Transition Type: 0x01=V-Fade, 0x03=X-Fade
%03x	DSK3 Fader Angle (0x0 to 0x200)
%03x	DSK3 Fade-to-Black Angle (0x0 to 0x200)
%03x	DSK4 Fader angle (0x0 to 0x200)
%03x	DSK4 Fade-to-Black Angle (0x0 to 0x200)

The information returned for DSK-3901 is:

Format	Field
%c	Cmd
%03x	DSK1 Fader Angle (0x0 to 0x200)
%03x	DSK1 Fade-to-Black Angle (0x0 to 0x200)
%03x	DSK2 Fader angle (0x0 to 0x200)
%03x	DSK2 Fade-to-Black Angle (0x0 to 0x200)
%02x	0 (Not used)

The information returned for LGK-3901 is:

Format	Field
%c	Cmd
%03x	DSK1 Fader Angle (0x0 to 0x200)
%03x	DSK1 Fade-to-Black Angle (0x0 to 0x200)
%03x	DSK2 Fader angle (0x0 to 0x200)
%03x	DSK2 Fade-to-Black Angle (0x0 to 0x200)
%02x	0 (Not used)
%03x	DSK3 Fader Angle (0x0 to 0x200)
%03x	DSK3 Fade-to-Black Angle (0x0 to 0x200)
%03x	DSK4 Fader angle (0x0 to 0x200)
%03x	DSK4 Fade-to-Black Angle (0x0 to 0x200)
%03x	DSK5 Fader angle (0x0 to 0x200)
%03x	DSK5 Fade-to-Black Angle (0x0 to 0x200)

Imagestore Intuition requires an extra “layer number” parameter and returns the status of a single layer.

CMD	Param_1
N	Layer No.

Imagestore Intuition example:

```
void EnquireVideoLayerStatus(int Layer)
{
    remote_send("N %x", Layer);
}
```

The information returned for Imagestore Intuition is:

Format	Field
%c	Cmd
%03x	Layer Fader Angle (0x0 to 0x200)
%03x	Layer Fade-to-Black Angle (0x0 to 0x200)
%03x	0
%03x	0
%02x	0

Fader Angle = 0x0 equates to “Keyer Off” or “FTB On”.

Fader Angle = 0x200 (512) equates to “Keyer On” or “FTB Off”.

Note: Transition types supported are not consistent between different products, so please check individual user manuals for details.

Set Preview Keyer Y1

This command is used to set layer fader angles on the Preview channel to a value which depends on the direction argument as follows:

0: Keyer Off (0)

1: Keyer On (512)

CMD	Param_1	Param_2
Y1	Layer No.	Direction

Example:

```
void SetPreviewKeyer(int Layer, int Dir)
{
    remote_send("Y1%1d %1d", Layer, Dir);
}
```

Set Preview Mixer Y3

This command sets which of the ABMixer channels must be shown on the preview output:

0: A channel

1: B channel

CMD	Param_1
Y3	ABMixer

Example:

```
void SetPreviewMixer(int ABMixer)
{
    remote_send("Y3%d", ABMixer);
}
```

Select Channel Input UF

Note: This command is only valid for LGK-3901 and DSK-3901

This command sets which of the SDI inputs will be shown on the program or preview channel:

CMD	Param_1	Param_2	Param_3
UF	%1x: Mixer Input 0 (fixed)	%1x: Channel 1 – Preview	%1x: Source 0 – Program In 4 – Fill 2

The current routing can be enquired using:

CMD	Param_1	Param_2
Y3	%1x: Mixer Input 0	%1x: Channel 0 – Program 1 – Preview

The information returned for LGK-3901 or DSK-3901 is:

Format	Field	Description
%c	Cmd	UF
%1x	Mixer Input	0
%1x	Channel	0 – Program 1 – Preview
%1x	Source	0 – Program In 4 – Fill 2 (only applicable for Preview channel)

Tally Commands

The Imagestore can be set into a mode where its state is fed back over automation without explicit request. This information is referred to as unsolicited tallies.

Enable Video Tallies Y6

This command enables or disables the Y6, Y9, YA, Yf and 3 video tallies for the serial/network port on which the command was received.

CMD	Param_1
Y6	%1x: Tally Enable 1=Enable, 0=Disable

Example:

```
void EnableVideoTallies(bool Enable)
{
    remote_send("Y6%1x", Enable);
}
```

For LGK-3901 and DSK-3901 this command has an additional optional parameter that enables or disables the ability to return tallies for commands that are sent to an Imagestore Intuition. When Intuition tally forward is enabled, tallies are returned by the Imagestore Intuition to the LGK-3901 or DSK-3901. See the user manual for more information.

CMD	Param_1	Param_2 (Optional)
Y6	%1x: Tally Enable 1=Enable, 0=Disable	%1x: Intuition Tally Forward Enable 1=Enable, 0=Disable

Note: Video tallies may also be enabled via the Set Passive Mode (Ya) command (page 247).

For all Imagestore products except Imagestore Intuition, this “Y61” command returns with:

CMD	Param1	Param2	Param3	Param4	Param5	Param6	Param7
Y6	%1x AB Mix Position	%1x DSK1 Keyer Position	%1x DSK 2 Keyer Position	%1x DSK 1 FTB Position	%1x DSK 2 FTB Position	%02x SQZ Preset/ Seq. No	%02x SQZ Mode/ Router Mode

Where the one hex digit format implies:

AB Mix Position	DSK Keyer Positions	DSK FTB Positions	Value
MIXER_AT_A	KEYER_OFF	FTB_ON	0
MIXER_AT_B	KEYER_ON	FTB_OFF	1
MIXER_INBETWEEN	KEYER_INBETWEEN	FTB_INBETWEEN	2

Note: The Y6 command returns information on the first two keying layers. Imagestore 750 keyer and FTB states for DSK3 and DSK4 are therefore broadcast via the 3 and Y9 tallies. LGK-3901 keyer and FTB states for DSK3, DSK4 and DSK5 are therefore broadcast via the 3 and Y9 tallies.

The Squeeze Preset/Sequence Number is the current squeeze preset or sequence number loaded. If the Squeeze is currently running (mid sequence) the preset number will be 0xFF.

For Imagestore 2, the Squeeze modes for the standard Squeezies are:

SQ_NONE	0
SQ_PIC_IN_PIC	1
SQ_REVEAL	2
SQ_STORED	3
SQ_REVEAL_MG	4
SQ_REVEAL_FG	5
SQ_STORED_MG	6
SQ_STORED_FG	7
SQ_B_OVER_A	8
SQ_BYPASS	9
SQ_MIX	10

For Imagestore 3 and Imagestore 300 fitted with a dual channel 2D/3D DVE the possible Router Modes are:

DVE_ROUTER_NONE	0
DVE_ROUTER_MIDGROUND_REVEAL	4
DVE_ROUTER_BYPASS	9
DVE_ROUTER_MIDGROUND_MIX_REVEAL	A
DVE_ROUTER_IS_MIGROUND_AB	B
DVE_ROUTER_IS_MIGROUND_BA	C
DVE_ROUTER_IS_MID_SQZ_A_MID	D
DVE_ROUTER_IS_MID_SQZ_MID_A	E
DVE_ROUTER_BEHIND	F

For Imagestore HD-TV the possible Router Modes are:

DVE_ROUTER_NONE	0
DVE_ROUTER_A_B_MIXER	1
DVE_ROUTER_A_AND_B_OVER_DSK1	4
DVE_ROUTER_BYPASS	9

For Imagestore 750 the possible Router Modes (PGM channel) are:

DVE_ROUTER_NONE	0
DVE_ROUTER_ACTIVE	4
DVE_ROUTER_BYPASS	9

For LGK-3901 and DSK-3901 the AB Mix Position, possible Router Modes and Sequence Number variables (parameters 1, 6 and 7) will be returned as zero since they are not supported

Note: Please refer to individual user manuals for further information on DVE modes.

The status of the Y6 tallies can be enquired using:

CMD
Y6

The information returned for the Imagestore 2/3, Imagestore 300[+], Imagestore HD-TV and Imagestore 750 is:

CMD	Param_1
Y6	%1x: Tally Enable 1=Enabled, 0=Disabled

The information returned for the LGK-3901 and DSK-3901 is:

CMD	Param_1	Param_2
Y6	%1x: Tally Enable 1=Enabled, 0=Disabled	%1x: Intuition Tally Forward Enable 1=Enabled, 0=Disabled

Request Wake-up Packet Y7

This command causes an unsolicited wake-up packet to be returned by an Imagestore.

CMD
Y7

Wake-up Packet Y7

The following unsolicited packet is sent by Imagestore on unit restart or on a request using above command. This allows the listening device know when an Imagestore has been restarted so that volatile settings (such as video/audio tallies and metering) can be reset. The packet also provides information about IP Address and serial number of the unit.

CMD	Param_1	Param_2
Y7	%04x Machine Serial Number	%08x Machine IP Address

Note: The Y7 command is normally only used by Presmaster.

Enable Audio Tallies Y8

This command enables or disables unsolicited audio tallies for the serial/network port on which the command was received.

CMD	Param_1
Y8	1=Enable, 0=Disable

Example:

```
void EnableAudioTallies(bool Enable)
{
    remote_send( "Y8%d" , Enable );
}
```

Note: Audio tallies may also be enabled via the Set Passive Mode (Ya) command (page 247).

Once enabled for a particular port, information regarding the current audio status is broadcast whenever audio state changes as follows:

CMD	Param_1	Param_2	Param_3	Param_4
Y8	%1x Audio AB Mix Position	%1x Audio Fade to Silence Position	%1x Voice Over 1 Position	%1x Voice Over 2 Position

Possible values for Audio AB Mix position are:

MIXER_AT_A	0
FADER_AT_B	1
FADER_INBETWEEN	2

Possible values for Audio Fade to Silence position are:

FADE_TO_SILENCE_ON	0
FADE_TO_SILENCE_OFF	1
FADE_TO_SILENCE_INBETWEEN	2

Possible values for the Voice Over positions are:

VOICE_OVER_OFF	0
VOICE_OVER_ON	1
VOICE_OVER_INBETWEEN	2

Note: For Imagestore 2, parameters 3 and 4 are “Voice Over 1 Pair 1” and “Voice Over 1 Pair 2” respectively.

The status of the Y8 tallies can also be enquired using:

CMD
Y8

This will return:

CMD	Param_1
Y8	%x 1=Enabled, 0=Disabled

Image Load Tallys Y9

The Y6 command (page 63) enables or disables unsolicited video tallies for the serial/network port on which the command was received. One of these video tallies is the Image Load Tally (Y9) which returns the filename of the image file that is currently loaded into a layer – via R0 (page 87)

Cmd	Param_1	Param_2
Y9	%1x Layer No.	%s Image File Name

As video tallies are enabled the Imagestore sends out Y9 tallys in bulk (one for each layer) to enumerate all of the currently loaded images. Subsequent changes to loaded images (via R0 or A commands) are then tallied on an individual basis.

Note: When a layer is unloaded the Image File Name will be equal to “> Empty <”.

Image Preload Tally **YA**

The Y6 command (page 63) enables or disables unsolicited video tallies for the serial/network port on which the command was received. One of these video tallies is the Image Preload Tally (YA) which returns the filename of the image that is currently pre-loaded into a layer – via R7 (page 93).

Cmd	Param_1	Param_2
YA	%1x Layer No.	%s Image File Name

YA tallies do not get sent in bulk as video tallies are enabled. However subsequent changes to preloaded images (via the R7 command) are tallied on an individual basis.

Enable Media Tallies **YB**

This command enables or disables media tallies for the serial/network port on which the command was received. Media tallies are used to track media files as they are added, deleted or modified on the file system

CMD	Param_1
YB	%06x Media Type (bitwise) 0x000001 = Images 0x000002 = Fonts 0x000004 = Sounds (not applicable for Intuition) 0x000008 = Mixes (not applicable for Intuition) 0x000010 = DVE sequences (not applicable for Intuition, LGK-3901 or DSK-3901) 0x000020 = Metadata presets (Imagestore 750 v2.0) 0x000040 = Configurations (Imagestore 750 v2.0, LGK-3901 and DSK-3901)

The command takes a bitwise parameter that defines what combination of media type tallies is required. If the file type parameter is set as 0x000000 then all media tallies will be disabled. In the following example only images and fonts are registered for.

```
void EnableMediaTalliesImagesFonts()
{
  remote_send( "YB000003" );
}
```

Once media tallies are enabled, unsolicited tallies are issued every time a matching media file is added, deleted or modified.

The format of the YB tally response is as follows:

CMD	Param_1	Param_2	Param_3
YB	%06x Media Type (not bitwise) As defined above	%1x 0x0 = Deleted 0x1 = Added 0x2 = Modified	%s Media File Name

The first parameter reports the media file type. This can not be a bitwise combination of media types since the tally is for a single media file.

The second parameter defines whether the image was added, deleted or modified (over-written) within the file system.

The third parameter defines the media file name for which the change took place.

Note: If media files are renamed then two tallies will be received; one for the deletion, and one for the addition (or modification).

The status of the YB tallies for any port can be enquired using:

CMD
YB

This will return:

CMD	Param_1
YB	%06x Tally mask (bitwise) as defined above

Heartbeat Tally YD

This command enables or disables heartbeat unsolicited tallies for the serial/network port on which the enable command was received. This command also sets the number of frames between each heart beat tally being sent.

CMD	Param_1	Param_2
YD	%1x 1=Enable, 0=Disable	%04x Frequency in frames 0x0001 to 0xFFFF

The format of the YD tally response is as follows:

CMD	Param_1	Param_2
YD	%1x 1=Enabled, 0=Disabled	%04x Frequency in frames

The status of the YD tally for any port can be enquired using:

CMD
YD

This will return:

CMD	Param_1
YD	%1x Tally Enable (1=Enabled, 0=Disabled)

Loaded Image Status Tally YO

This command enables or disables Loaded Image Status tallies for the serial/network port on which the command was received.

CMD	Param_1
YO	1=Enable, 0=Disable

Any subsequent changes in state for the list of parameters returned by the Enquire Loaded Image Status command “O” (page 45) will result in an unsolicited tally being sent with exactly the same form as the normal response.

Fade-To-Black State Tally Yf

The Y6 command (page 63) enables or disables unsolicited video tallies for the serial/network port on which the command was received. This includes the Yf tally which returns keyer Fade-to-Black position information for each layer as state changes (unsolicited).

CMD	Param_1	Param_2
Yf	%1x Layer No.	%1x Fade-to-Black Position

Possible values for Keyer Fade-to-Black Position are:

FTB_ON	0
FTB_OFF	1
FTB_INBETWEEN	2

Text Crawl Count Tally Yg

Note: This command is only enabled if either the video tallies are enabled (Y61) or the device is set to operate in passive mode [Ya1].

This tally is sent by Imagestore Intuition every time a text crawl completes for text that been setup to send this tally had cycled. The Yg command is enabled using the Y6 or the Ya command. This command enables or disables the text crawl count tally and is used in conjunction with the Zh command. For a tally to be issued for any object, the 'Tally' checkbox should be set within Intuition Builder.

For Imagestore Intuition this will return the following tally every time that a text crawl completes:

CMD	Param_1	Param_2	Param_3	Param_4
Yg	%1x Layer No.	%02x Box No.	%1x Final Crawl Flag	%04x Crawl Count

The final crawl flag is 0x1 if the crawl just completed was the final one.

Animation/Clip Cycle Tally Yj

Note: This command is only valid for Imagestore Intuition. This command is only enabled if either the video tallies are enabled (Y61) or the Imagestore Intuition is set to operate in passive mode [Ya1].

This tally is sent by Intuition every time a cell animation/clip that been setup to send this tally had cycled. The tally is sent at the end of each cycle and is enabled using the Y6 or the Ya command. For a tally to be issued for any object, the 'Tally' checkbox should be set within Intuition Builder.

For Imagestore Intuition this will return the following tally every time that a clip or animation cycle completes:

CMD	Param_1	Param_2	Param_3	Param_4
Yj	%1x Layer No.	%02x Object No.	%1x Final Cycle Flag	%04x Crawl Count

The "Final Cycle" flag in the tally command works differently for each loop mode, for example for layer 2, object 4:

```
// Single shot: Send single tally when the animation/clip ends
Yj2 04 1 0001

// Cycle: Send tally each time the animation/clip cycles,
// giving the current crawl count.
Yj2 04 0 ?????

// In-Loop-Out: Send single tally when the whole
// animation/clip has completed, and does not tally on inner
// loops.
Yj2 04 1 0001

// Linear control: Send single tally every time a move
// completes, giving the current crawl count.
Yj2 04 1 ?????

// Ping pong: Send tally at either end of the animation,
// giving the current crawl count.
Yj2 04 0 ?????
```

Keyer Position Tally

3

Note: The Y6 tally returns keyer position state for DSK1 and DSK2. The 3 tally is therefore only implemented on products with more than two keying layers – (Intuition, Imagestore 750 and LGK-3901).

The Y6 command (page 63) enables or disables unsolicited video tallies for the serial/network port on which the command was received. This includes the 3 tally which returns keyer position information for each layer as state changes (unsolicited).

Cmd	Param_1	Param_2
3	%1x Layer No.	%1x Keyer Position

Possible values for Keyer Position are:

KEYER_OFF	0
KEYER_ON	1
KEYER_INBETWEEN	2

Audio Commands (old)

Note: The commands in this section are largely deprecated and ONLY apply to Imagestore 2 with 'Easysound' option disabled.
More recent Imagestore products utilise the Easysound audio model which is described on page 159.

Audio commands are applicable to systems that have the Easysound audio mixer layer fitted. The audio system is viewed as a matrix with variable gain cross-points. The mixes output from the matrix are then patched to physical outputs, each of which has a master gain (fade to silence) control. Each of the mix matrix inputs is also available at the patch layer to allow pre-mix monitoring. See the diagram on page 83.

The inputs into the matrix may be selected from embedded audio or from physical AES inputs. The outputs from the patcher stage are available as AES and optionally as audio embedded into the program SDI output.

In practice there are physical constraints which limit the maximum number of logical outputs, the maximum number of physical outputs and the maximum number of inputs which may simultaneously contribute to any given logical output. This depends on the hardware purchased.

Inputs may be treated as mono or stereo. Inputs may be phase inverted or silenced.

Mixer outputs are controlled by sending sets of gain, input and time values for each input fader to be changed. Mix matrix outputs can be specified as left, right or stereo.

Audio Commands Table

The table below shows the commands used to control the audio layer. Each entry in the table shows a parameter to be put in the command string. These parameters are not separated by spaces as they are all of fixed length. The values for each parameter are of set types that are described in the section starting page 80.

The number ranges for most parameters are zero-based. The audio General Purpose Interface (GPI) programming script controls the same Easysound audio layer features and functions as the automation protocol, but uses unity-based parameters. This is an important consideration for those planning to control Oxtel units using both audio GPI script files and serial automation control.

Audio Command	ID	Param_1	Param_2	Param_3
Matrix input pre-select	a	Matrix Input	Pre select Source	
		path_no	phys_io	
Set input mode	b	Matrix Input	Input Mode	
		path_no	io_mode	
Set input pan	c	Matrix Input	Pan Value	
		path_no	pan_val	
Set mix	d	Matrix Output	Input Level Block	
		path_no	mix_blk	
Set mix mode	e	Matrix Output	Output Mode	
		path_no	io_mode	
Set output patch	f	Patch Output	Patch Input	Select matrix input or output
		path_no	path_no	patch_source
Set output gain	g	Patch Output	Gain	Duration
		path_no	gain_val	duration
Set embedded output	h	Patch Output	Embedded channel	
		path_no	phys_io	

Matrix Input Preselect a

Note: This command is largely deprecated.

This command selects which physical input is pre-selected to the matrix input. The parameters are the matrix input and the pre-select input.

Example:

```
void SetInputPath(UINT8 byPhysio, UINT8 byPathNo)
{
    remote_send( "a%c%c", byPathNo+0x20,
                byPhysIo+0x20 );
}
```

Set Input Mode b

Note: This command is largely deprecated.

This command sets the input to the matrix to be either mono or stereo and also sets the input to carry one of the input unmodified, the input replaced by silence or the input phase inverted. This takes two parameters: the matrix input and the input mode. When an input is set to stereo the command will affect both channels. When set to mono it will only affect the indicated channel. Setting the mode to stereo or mono affects the other channel in the channel pair.

Example:

```
void SetInputMode(UINT8 byInPath, UINT8 byMonoBit,
                  UINT8 byMode)
{
    UINT8 byIOMode;
    byIOMode = 0x20 + ((byMode << 1) | byMonoBit);
    remote_send( "b%c%c", byInPath+0x20, byIOMode );
}
```

Set Input Pan c

Note: This command is largely deprecated.

This command sets the proportion of each input passed in a stereo input or the position of a mono input into a stereo mix. This takes an input and a pan value.

Set Mix d

Note: This command is largely deprecated.

This command sets one or more contributions to a given mix. All inputs that are not explicitly mentioned in the command are unchanged. At least one mix_block must follow the matrix output value.

Example:

```
void SetMix(UINT8 byOutPath, UINT8 byInPath,
            UINT8 byGain, UINT8 byDuration)
{
    remote_send("d%c%c%02x%02x", byOutPath+0x20,
                byInPath+0x20, byGain, byDuration);
}
```

Set Mix Mode e

Note: This command is largely deprecated.

This command sets the mode of a mix matrix output to either mono or stereo. The parameters are the output and the mode.

Set Output Patch f

Note: This command is largely deprecated.

This command selects which signal is patched to a given patch panel output. One source may feed many outputs. A separate parameter is used to say if the value is an input to the matrix or an output from it.

Example:

```
void SetOutputPatch(UINT8 byOutPathNo,
                     UINT8 byInPathNo,
                     UINT8 byPatchSource)
{
    remote_send("f%c%c%c", byOutPathNo+0x20,
```

```
        byInPathNo+0x20, byPatchSource+0x20);  
    }
```

Set Output Gain g

Note: This command is largely deprecated.

This command sets the fade to silence fader associated with a patch panel output. The duration value sets the number of frames over which the action is to take place.

Example:

```
void SetOutputGain(UINT8 byPathNo, UINT8 byGain,  
                    UINT8 byDuration)  
{  
    remote_send( "g%c%02x%02x", byPathNo+0x20,  
                byGain, byDuration);  
}
```

Set Embedded Output h

Note: This command is largely deprecated.

This sets which groups the audio outputs are embedded into. The AES output is always available and need not be explicitly as it must be when setting the mode of an input.

Parameter Ranges & Meanings

This section explains the format of the parameters used within the audio commands listed in the table on page 76.

phys_io

This is used to pre-select the source taken into a matrix input or the destination to which an output is sent. Values can be selected from embedded channel pairs or from separate physical AES streams. Value 0x20 selects embedded channel pair 1. 0x27 selects channel pair 8. 0x28 selects the associated AES stream.

Note: On IS2, there is no audio de-embedding function on the SDI Fill input path number.

This is the number of the input or output as though it was a mono mixer. Alternatively, the input may be thought of as the stereo input number followed by a one bit left/right identifier, with Left being numerically lower than Right.

The value is computed as 0x20 + path number:

0	0	1	0	0	0	0	0
+ 0	0	n	n	n	n	n	p

nnnnn = STEREO input number

p = 0 Left

p = 1 Right

0x20 -> path 0 pair 1 LEFT

0x21 -> path 1 pair 1 RIGHT

0x22 -> path 2 pair 2 LEFT

0x23 -> path 3 pair 2 RIGHT

0x24 -> path 4 pair 3 LEFT

0x25 -> path 5 pair 3 RIGHT

0x26 -> path 6 pair 4 LEFT

0x27 -> path 7 pair 4 RIGHT

0x28 -> path 8 pair 5 LEFT

0x29 -> path 9 pair 5 RIGHT

0x2a -> path 10 pair 6 LEFT

0x2b -> path 11 pair 6 RIGHT

io_mode

I/O mode defines the way that commands for inputs and outputs are interpreted. For inputs this controls whether the input is a separate mono input or is treated as part of a stereo pair. It also controls whether the signal is phase inverted or replaced by silence. For outputs, only the mono/stereo bit is significant.

0 0 1 0 0 a a x	
x = 0	STEREO
x = 1	MONO
aa = 00	normal
aa = 01	inverted
aa = 10	silence
aa = 11	undefined

Setting either part of a pair to mono or stereo forces the other part to follow suit.

gain_val

Gain is specified in 1/2 dB steps with 1 meaning -99.5dB. 0 is a special case and means infinite attenuation. The maximum gain which may be specified is +28dB. This value is two bytes long comprising two hex digits forming an unsigned byte. Thus, 00 implies infinite attenuation, 01 implies -99.5dB, C7 implies 0dB (unity gain), and FF implies +28dB.

duration

The duration is used to set the number of frames over which a gain change will take place. It is encoded as one hexadecimal value of two bytes width. The two bytes are further encoded as ASCII characters.

For example: To specify a duration value of 12 (decimal), which is 0C in hex:

12 (decimal)	->	0C (hex)	->	0x30 0x43
				"0" "C"

mix_blk

A mix_blk is used to set one input component of a mix. It consists of a sequence of three values contained in 5 bytes. The first value is one byte

wide and is a **path_no**. The remaining 4 bytes are encoded as two hexadecimal values each of two bytes width.

The first value selects the contributing input, the second value is the required gain, and the third value is the duration, in video frames, over which the transition is to take place.

Example:

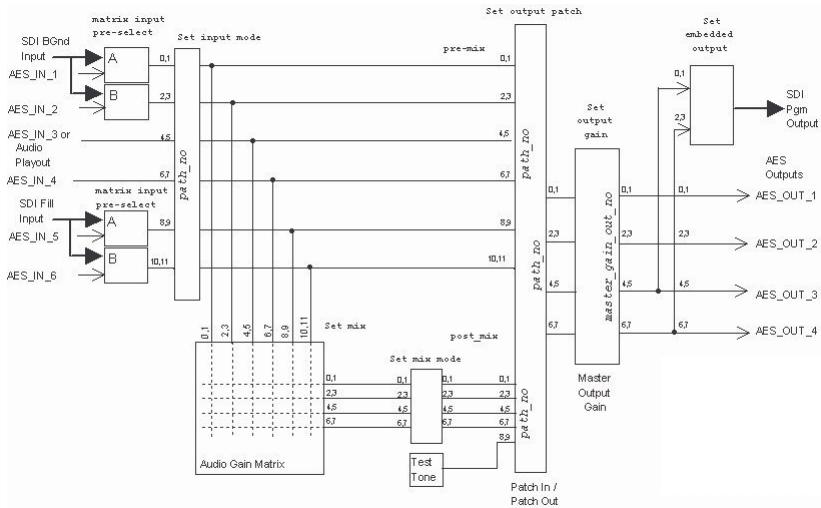
```
0x23 0x36 0x30 0x32 0x30

0x23 > path 3 if MONO, 3 and 4 if STEREO
0x36 0x30 -> 60(hex) -> 96 -> (96/2) - 99.5
= -51.5db
0x32 0x30 -> 20(hex) -> 32 -> take 32 video
frames.
```

patch_source

This value is set to 0x20 if the patcher is to monitor the matrix inputs and it is set to 0x21 to monitor the matrix outputs.

Easysound Automation Block Diagram



Extended Commands

Extended commands are two-byte commands with parameters following immediately on from the second command code byte. The first command byte identifies a command group, with the second byte identifying the specific command within the group.

All extended commands use fixed-width hexadecimal representations for integer parameters.

All string parameters are variable length, and are therefore grouped at the end of the packet. If multiple string parameters are required, they are delimited by vertical bar characters ('|') which therefore cannot appear in the parameter.

String parameters can contain 'escape' sequences to include characters which cannot otherwise be transmitted using the remote protocol – especially colon '\3A', semicolon '\3B' and vertical bar '\7C'. These are handled using C-style escape sequences using a backslash character followed by two hexadecimal bytes to form the code for a single byte. Note that backslash itself must therefore be transmitted as '\5C'

This code converts normal ASCII strings into a format suitable for sending.

```
char * put_remote_str(char *instr, char *outstr)
{
    char c;
    for(;;)
    {
        c = *instr++;
        if (c == 0)
        {
            *outstr = 0;
            return outstr;
        }
        if ((c == ';') || (c == ':') || (c == '|') || (c == '\\\\'))
        {
            *outstr++ = '\\';
            sprintf(outstr, "%02x", c);
        }
        else
        {
            *outstr++ = c;
        }
    }
}
```

{}

Some floating-point parameters are converted to 16.16 fixed-point integer representation, and sent as an 8-character hex field, so 3.5 would be sent as 00038000.

```
long oxfixed(float f)
{
    return f * 0x10000;
}
...
send_command("ZZ%08x%08x", oxfox(PI), oxfixed(1.23456));
...
```

Filename-Based Commands

The following commands are provided to allow access to images stored on the Oxtel Series products. These products support alphanumeric filenames (including the file extension) of up to 39 characters.

Generally speaking the file-systems on Oxtel Series products are case insensitive. However this is not the case for Imagestore HD-TV, Imagestore 300, and Intuition products that utilise the EXT2 file-system. It is therefore recommended that filenames are never duplicated with different cases, and that automation commands that pass filenames do maintain case.

It is recommended that filename extensions be included within any automation commands. This will ensure that the correct file is always utilised.

It is recommended that filenames contain only the characters A-Z, a-z, 0-9, and underscore.

Load Image **R0**

This command loads an image, animation, Easytext file or clock onto the specified layer. Unlike the 8 command, the image is loaded by name, rather than by number. If the layer is currently faded up, the image will be seen to wipe down the screen. The R0 command supersedes the command 8 (which remains supported). If a non-existent filename is supplied as Param_2, the current image will be removed and no new image will replace it.

The following file types are typically loaded via the R0 command⁴:

- Still images (.OXT)
- Animations / Bugclocks / Easytext (.OXA)
- Intuition Templates (.TEM) – Intuition only
- MPEG video clips (.CLP) – Imagestore 2 only

Within Imagestore Intuition, several different formats of MPEG video clips can be included within a template, but not loaded directly into Intuition layers.

CMD	Param_1	Param_2
R0	%1x: Layer No.	Filename

⁴ Please see individual User Manuals for exact details on file-types supported

Example 1:

```
void LoadEmergencySlide(int Layer)
{
    remote_send("R0%lx%s", Layer, "Emergency.oxa");
}
```

Example 2:

```
void LoadNumberedImage(int Layer, int num)
{
    //Equivalent to remote_send("8%x %x", Layer, num);
    remote_send("R0%lxV%03d", Layer, num);
}
```

When no filename is specified the R0 command alternatively responds with the name of the currently loaded file. This is useful when enquiring which file is loaded into a layer.

CMD	Param_1
R0	%1x: Layer No.

Example 3:

```
void EnquireFileLoaded(int Layer)
{
    remote_send("R0%lx", Layer);
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	R0
%1x	Layer	Layer No.
%s	Filename	Confirmation of media filename, or ">Empty<" if no file is loaded. External fill and key is shown by ">F1+K1<", ">F2+K2<", etc.

Save Image R1

This saves a previously loaded or grabbed image to disk. The Image is saved with all its currently selected keying, masking and positioning attributes.

The mask flag is only used in Imagestore 2/3. If the Mask Flag is 0, then the entire image is saved. If the mask flag is 1, then only the area of the image within the current mask rectangle is saved, thus reducing disk space and increasing load times.

The requirement to save disk space is less important on Imagestore 300, Imagestore HD-TV, Imagestore 750 and LGK-3901. For this reason the mask flag has no effect on these systems and the complete image is always preserved to allow a crop to be reversed at a later date.

The filename and title are string parameters – the title is a simple text field which can provide additional information about an image. Titles are not required to be unique.

This command supersedes and extends the 9 and I commands (which remain supported)

CMD	Param_1	Param_2	Param_3	Param_4
R1	%1x: Layer No.	%1x: Mask Flag: 0 = Full Image, 1 = Masked area	Filename	Title

Example: Save image

```
void SaveImage(int Layer, bool Masked,
               char* filename, char* title)
{
    remote_send("R1%1x%1x%s|%s", Layer, Masked,
                filename, title);
}
```

Erase Image R2

This command erases a specified media file from the unit's internal disk. This command supersedes and extends the Q command (which remains supported)

CMD	Param_1
R2	Filename

Example:

```
void EraseEmergencySlide()
{
    remote_send( "R2%s" , "Emergency.oxa" );
}
```

Enquire File Info R3

This command retrieves whether a specified media file exists (or not) on the unit's internal disk. This command supersedes and extends the 'P' command (which remain supported).

Note: To query audio files use the R8 command.

CMD	Param_1
R3	Filename

Example:

```
void EnquireEmergencySlide()
{
    remote_send( "R3%s" , "Emergency.oxa" );
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	R3
%1x	File exists	1 = File exists, 0 = File not present
%s	Filename	Confirmation of filename

Query First File R4

This command is used to find the name of the first file within a specific directory. To retrieve the names of subsequent files use the 'R5' command. The order of the filenames retrieved is generally alphabetical⁵.

CMD	Param_1
R4	Folder name alias

⁵ May not be true for older products.

The folder name alias is a string which can be one of the following

- \$VIDEO: Image directory
- \$AUDIO: Audio directory
- \$FONTS: Font directory

Example:

```
void QueryFirstFile ()
{
    remote_send( "R4%s" , "$VIDEO" );
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	R4
%1x	End of directory	1 = End of directory reached, 0 = Otherwise.
%s	Filename	Name of the file, or “unknown” if directory does not exist, or “end” if end of directory reached

Query Subsequent File R5

This command is used in conjunction with R4 to request all the files from a specific directory on the Imagestore.

The order of the filenames retrieved is generally alphabetical.

CMD	Param_1
R5	Folder name alias

The folder name alias is a string which can be one of the following

- \$VIDEO: Image directory
- \$AUDIO: Audio directory
- \$FONTS: Font directory

The information returned is in exactly the same format as for the ‘R4’ command, but with ‘R5’ as the command code.

Format	Field	Description
%c%c	Cmd	R5
%01x	End of directory	1 = End of directory reached, 0 = Otherwise.
%s	Filename	Name of the file, or "unknown" if directory does not exist, or "end" if end of directory reached

Enquire Extended File Info R6

This command retrieves information about a specified image, animation or clock from the internal disk of the Imagestore device.

CMD	Param_1
R6	Filename

Example:

```
void EnquireEmergencySlide()
{
    remote_send( "R6%s" , "Emergency.oxa" );
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	R6
%01x	File exists	1 = File exists, 0 = File not present
%03x	X Position	Horizontal position of image
%03x	Y Position	Vertical position of image
%03x	Width	Width of image
%03x	Height	Height of image
%03x	Clip	Clip value of the image
%03x	Gain	Gain value of the image
%03x	Transparency	Transparency value of the image
%02x	Image Type	1=Still, 2=Animation, 4=Easytext, 5=Easytext crawl/roll, , 8=Clock
%04x	Number of frames	Number of frames if it is an animation
%01x	Animation Mode	Animation play mode if it is an animation: 0=Cycle, 1=Single shot, 2=In loop out, 3=Linear control, 4=Ping pong, 5=Multi loop
%02x	Load Time	Estimated time to load image in seconds
%01x	Associated Audio	1 if image has associated audio file.
%s	Filename	Confirmation of filename

Example of parsing the information returned (using C).

```
sscanf(&buff[1],
    "R6%01x%03x%03x%03x%03x%03x%03x%02x%04x%01x%02x%01x%s",
    &Info->file_exists, &Info->xpos, &Info->ypos,
    &Info->width, &Info->height, &Info->clip, &Info->gain,
    &Info->trans, &Info->image_type, &Info->no_frames,
    &Info->anim_mode, &Info->load_time,
    &Info->associated_audio, Info->FileName);
```

Preload Image R7

This command is used to preload an image, animation or clock for the specified layer. This copies the media file into an off-screen (free) area of store memory. When a subsequent load command (R0) is issued, the preloaded image name is checked against the image requested. If they match, then the preloaded image is immediately selected onto that layer. This enables on-air media to be swapped with new media without any load delay.

If a different file is requested via 'Load', then the preloaded image is cleared.

If no extension is specified, the Imagestore will try both .OXA and .OXT.

Note: On Imagestore 2 and earlier products, there is a single process which handles all file loads, saves, deletes and preloads in strict order of reception. In other words, if a load is in progress on layer 0, load (or save, etc.) requests on another layer will be deferred until the first load completes.

This command (as with other filename-based commands) will return an ACK as soon as the request has been queued, rather than waiting for its completion.

Note: For long animations, please be aware that you may not be able to hold both a pre-loaded image and the current on-air image in the store memory available. In this case, the preload will fail. Larger video memory will address this problem.

Note that the progress of the preload can be monitored by checking the 'disk busy' bit returned by the **Enquire loaded image status 'O'** command as part of the key type field.

One disadvantage of using R7 is that the new media can not be previewed prior to it being shown on-air (assuming the Program keyer remains cut up).

CMD	Param_1	Param_2
R7	%1x: Layer No.	Filename

Example 1:

```
void PreloadEmergencySlide(int Layer)
{
    remote_send( "R7%1x%s" , Layer , "Emergency.oxa" );
}
```

Enquire Audio File Info R8

This command retrieves whether a specified audio file (OXE, OXW or WAV) exists on the Imagestore unit's hard disk. This command supersedes and extends the 'Q' command (which remains supported).

Note: OXE files are not supported by Imagestore 2/3.

CMD	Param_1
R8	Filename

Example:

```
void EnquireAudioFile( )
{
    remote_send( "R8%s" , "song.wav" );
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	R8
%1x	File exists	1 = File exists, 0 = File not present
%s	Filename	Confirmation of filename

Emergency to Air R9

This command automatically displays a still picture which must be named V000.OXT. This appears on the most downstream keying layer so that it appears over the background video and any upstream graphics and/or DVE. When this automation command is sent with ON the current active layer state is saved and faded down and the default image (V000.OXT) is then faded up. When the automation command is sent with OFF the default image is faded down and the previous state restored.

CMD	Param_1
R9	0=Off, 1=On

Example:

```
void EmergencyToAir(bool onOff)
{
    remote_send( "R9%1x" , onOff );
}
```

Image Load Mode Rm

This command sets the current image load mode for the specified layer. Mode can be one of the following:

0: Clean Load:

- Old image cut down
- New image loads
- New image cuts up

1: Cut Load:

- Old image held while new image loads
- Cut to new image

CMD	Param_1	Param_2
Rm	Layer No.	Mode

Example 1:

```
void ImageLoadMode(int Layer, int Mode)
{
    remote_send( "Rm%1x%x" , Layer, Mode );
}
```

When no mode is specified the Rm command alternatively responds with the current mode for the layer.

CMD	Param_1
Rm	Layer No.

Example 2:

```
void EnquireImageLoadMode(int Layer)
{
    remote_send( "Rm%lx", Layer );
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	Rm
%1x	Layer	Layer No.
%x	Mode	Current mode

Validate Template RA

This command allows the media references made within Intuition templates to be validated.

Intuition template files typically reference many other media files including OXT images, and OXI animations. It is therefore important to know that all of these references are valid before bringing the template to air.

CMD	Param_1
RA	Filename

The information returned is:

Format	Field	Description
%c%c	Cmd	RA
%s	Filename	Template filename, ending with pipe separator
%01x	Template exists	1=Template exists 0=Template missing
%04x	Number of missing files	How many files in the subsequent list
%s %s. . . %s	Missing media	Piped list of missing media filenames

Examples of information returned from the automation enquire of the contents of MyTemplate.tem using the command "RAMyTemplate.tem"

- 1) The template does not exist: RAMyTemplate.tem|00000

- 2) The template exists, and all related media is present on the unit:
RAMyTemplate.teml10000
- 3) The template exists, but one related media file is missing (test1.oxi):
RAMyTemplate.teml10001\test1.oxi
- 4) The template exists, and media files test1.oxi -> test256.oxi are missing:
RAMyTemplate.teml10100\test1.oxi\test2.oxi\test3.oxi\.....\test256.oxi

After any missing files have been replaced, the RA enquire command should be sent one more time to confirm that all related media is now present.

Note: The RA command is only supported by Imagestore Intuition[+] and Intuition XG.

Get Audio File Duration RB

This command allows the duration of an audio file to be enquired.

CMD	Param_1
RB	Filename

The information returned for a valid audio file is:

Format	Field	Description
%c%c	Cmd	RB
%s	Filename	Audio filename
%02d	Hours	Number of hours
%02d	Minutes	Number of minutes
%02d	Seconds	Number of seconds
%02d	Frames	Number of frames

Animation Commands

The animation control commands are used to control the playout of animation sequences which have been loaded into the playout memory. They can be used to trigger 'single-shot' animations, and make other types of animation.

On Intuition units, animations occupy a template box rather than a full layer. These commands imply a box number of 0. See also the Zf command on page 232. Note that these commands are not supported on the DSK-3901.

Start Animation **S0**

This command starts or restarts an animation playing out. The animation starts playing from the current field. To force playout from the beginning, set the current field before issuing this command.

CMD	Param_1
S0	%1x : Layer No.

Example:

```
void StartAnimation(int layer)
{
    remote_send("S0%01x",layer);
}
```

Stop Animation **S1**

This command halts an animation. If the *complete cycle* flag is set, the animation completes its cycle before halting. In some animations, this may cause a special *stop section* of the animation to execute. If the immediate flag is set, then animation halts immediately at the specified frame. Animations always halt showing two frames to form a field. If there is much inter-field motion, the image will flicker.

CMD	Param_1	Param_2
S1	%1x: Layer No.	%1x: 0 = Complete cycle, then stop. 1 = Immediate stop

Example:

```
void StopAnimationImmediate(int layer)
{
    remote_send("S1%01x1",layer);
```

Select Animation Frame S2

This command sends the animation to a specified frame. If the animation is running the animation will continue to run from the specified frame. If the animation is halted, then the specified frame will be displayed but the animation will remain halted.

CMD	Param_1	Param_2
S2	%1x: Layer No.	%4x: Field No.

Example:

```
void JumpToField(int layer, int field)
{
    remote_send("S2%01x%04x",layer,field);
```

Run Pan Effect S3

This command runs a pan effect on an animation over a specified number of frames. Speed is measured in pixels/lines per field.

CMD	Param_1	Param_2	Param_2
S3	%01x: Layer No.	%03x: Effect No.	%03x: Speed

Effect	Effect no. fwd	Effect no. rev
Pan on from left	1	21
Pan on from right	2	22
Pan on from top	3	23
Pan on from bottom	4	24
Wipe on from left	11	31
Wipe on from right	12	32
Wipe on from top	13	33
Wipe on from bottom	14	34

Wipe on from centre (barn door)	15	35
Wipe on from centre (blinds)	16	36

Note: The S3 command is only supported by Imagestore 2.

Restart Animation S4

This command always restarts an animation playing out from the beginning. This command is a convenient alternative to the S0 command when the animation has to always start from the first field.

CMD	Param_1
S4	%1x : Layer No.

Example:

```
void ReStartAnimation(int layer)
{
    remote_send( "S4%01x" ,layer );
}
```


Bugclock Commands

The Bugclock commands allow the control of the timer features of a Bugclock image, for units where Bugclocks are supported. Clocks can be loaded, positioned, keyed, etc. using the standard commands for stills and animations (R0, G, 3, etc.).

Stop Timer **T0**

This command pauses a Bugclock timer on the specified layer.

CMD	Param_1
T0	%01x : Layer No.

Example:

```
void StopTimer(int layer)
{
    remote_send("T0%01x", layer);
```

Start Timer **T1**

This command allows a previously paused timer to resume.

CMD	Param_1
T1	%01x : Layer No.

Example:

```
void StartTimer(int layer)
{
    remote_send("T1%01x", layer);
```

Reset Timer T2

This command resets the Bugclock timer on the specified layer to its default time value. For count-up timers, this is typically zero.

CMD	Param_1
T2	%01x : Layer No.

Example:

```
void ResetTimer(int layer)
{
    remote_send("T2%01x", layer);
```

Set Timer to Value T3

This command sets a Bugclock timer to the specified value. Time-of-day clocks and time-of-day countdowns are not affected.

CMD	Param_1	Param_2	Param_3	Param_4
T3	%01x : Layer No.	%02x : Hour	%02x : Minute	%02x : Second

Example:

```
void SetTimer(int layer,int hour,int min,int sec);
{
    remote_send("T3%01x%02x%02x%02x", layer, hour,
                min, sec);
```

AB Mixer Commands

These commands control the AB mixer functionality of the Imagestore. For older products, SDI inputs A and B provide fixed inputs for the AB mixer. However newer products with an input router (Imagestore HD-TV and Imagestore 750) are able to route any of the available SDI inputs into the AB mixer.

The commands are not supported by the following LOGO products:

- Imagestore 300+ LOGO
- Imagestore HD-TV LOGO
- Imagestore 750 LOGO

Cut to A U0

This command cuts immediately to the A input.

CMD
U0

Example:

```
void CutToA()
{
    remote_send( "U0" );
}
```

Cut to B U1

This command cuts immediately to the B input.

CMD
U1

Fade to A U2

This command fades to the A input over a specified number of fields.

CMD	Param_1
U2	%03x: Fields

Example:

```
void FadeToA(int fields)
{
    remote_send("U2%03x", field);
}
```

Fade to B **U3**

This command fades to the B input over a specified number of fields.

CMD	Param_1
U3	%03x: Fields

Cut AB **U4**

This command cuts to the opposite input from the one currently visible

CMD
U4

X-Fade AB **U5**

This command fades to the opposite input to the one currently visible over specified number of fields.

CMD	Param_1
U5	%03x: Fields

Set Transition Type **U6**

This command selects the transition type to be used in commands U2, U3, U5 & U9.

The options are between mix, vertical wipe, horizontal wipe and V-Fade and X-Fade.

CMD	Param_1
U6	%2x: Transition Type

The following transition types are supported:

- 01: V-Fade
- 03: X-Fade
- 05: Cut
- 06: Wipe L to R
- 07: Wipe T to B
- 10: Wipe R to L
- 11: Wipe B to T

Note: **Wipes are NOT supported on Imagestore HD-TV, Imagestore 750, LGK-3901 or DSK-3901.**

Example:

```
void SelectCrossFade( )  
{  
    remote_send( "U603" );  
}
```

Note: **To set a colour field for the mid-point of a V-Fade, use the UD command.**

Wipe Softness Adjustment U7

This command sets the edge softness for Horizontal and Vertical wipes. Softness is measured in arbitrary units.

CMD	Param_1
U7	%2x: Wipe softness – 0..255

Example:

```
void SetWipeSoftness(int softness)  
{  
    remote_send( "U7%02x" , softness );  
}
```

V-Fade AB U8

This command causes the AB mixer to fade from the current input to the other via the V-fade colour (see UD) over specified number of fields. A separate duration for the *up* and *down* phases is supported if asymmetric V-fades are required.

CMD	Param_1	Param_2
U8	%03x: Fields for 'down' section of fade.	%03x: Fields for 'up' section - allows for asymmetric V-fades

Set Absolute Mix U9

This command sets the AB mix position to the specified absolute value. The absolute mix value can range from 0 (A=100%, B=0%) to 512 (A=0%, B=100%)

CMD	Param_1
U9	%03x: Value

Asymmetric Transition UA

This command performs V-fade, X-fade, cut or wipe (according to current Transition Mode – set with U6 command) to the destination specified.

For X-fades, V-fades and Wipes, the total duration is the down section of the fade (Param_1). For cuts, the duration is ignored.

CMD	Param_1	Param_1	Param_2
UA	%01x: Destination: 0 = A input 1 = B input	%03x: Fields for 'down' section of fade	%03x: Fields for 'up' section - allows for asymmetric fades

Transition Complete UB

Issued by Presmaster on completion of an AB mix/wipe transition to take input back to 'A'.

Similar to 'Cut to A', but includes compensating delays for Squeezy, etc.

CMD
UB

Fade to Specified Position UC

Transitions the AB Mixer to the specified position over the given number of fields. The position can be anywhere from 0 (A) to 0x200 (B) and all values in between. This can be used for U-Fades, by selecting V-Fade transition type (U2 above), then sending for example UC100050 (transition to mid-point of V-Fade) on Take pressed, then UC200050 (transition to B) on take released.

CMD	Param_1	Param_1
UC	%03x: Destination: 0x000 = A input, 0x200 = B input	%03x: Fields for duration fade.

Set V-Fade Colour UD

This command defines the colour that the AB Mixer V-fade will go through. The default is black.

CMD	Param_2	Param_3	Param_4
UD	Red: %02x	Green: %02x	Blue: %02x

For Imagestore 2/3 the R, G and B values must be between 0 and 255 (0x00 to 0xFF hexadecimal).

For Imagestore 300[+], Imagestore HD-TV and Imagestore 750 the R, G, and B values must be between 0 and 100 (0x0 to 0x64 hexadecimal) since they are all stored as percentage values, as viewed on the front panel.

Select Mixer UE

This command is used to route video sources to the video mixer. The mixer parameter can be set as 0 to select mixer input A, or 1 to select mixer input B. The source parameter depends on how many video source inputs are connected (maximum of 6 for Imagestore HD-TV, maximum of 8 for Imagestore 750).

- 0: Input A
- 1: Input B
- 2: Fill 1
- 3: Key 1
- 4: Fill 2
- 5: Key 2
- 6: Input C (Imagestore 750 only)
- 7: Input D (Imagestore 750 only)

CMD	Param_1	Param_2
UE	Mixer	Source

Example 1:

```
void SelectMixer(int Mixer, int Source)
{
    remote_send("UE%1x %1x", Mixer, Source);
}
```

The command UE alternatively responds with the current source for the mixer if no source is specified.

CMD	Param_1
UE	Mixer

Example 2:

```
void GetMixer(int Mixer)
{
    remote_send("UE%1x", Mixer);
}
```

The information returned is:

Format	Field	Description
%e%c	Cmd	UE
%1x	Mixer	Mixer selector
%1x	Source	Current source

Enquire Mix Mode Ua

This command returns the status of various mix parameters, such as transition type, mix rate, wipe softness, current mix angle and v-fade colour.

CMD
Ua

The Status report returned is.

Format	Field	Description
%c%c	Cmd	Ua
%02x	Transition type	<u>Imagestore 300[+1] / HD-TV / 750</u> Return type is consistent with the U6 command <u>Imagestore 2 / 3</u> Cut = 0 X-Fade = 1 V-Fade = 2 Wipe left to right = 3 Wipe right to left = 4 Wipe top to bottom = 5 Wipe bottom to top = 6
%03x	AB mix rate	Value in fields
%03x	Wipe softness	Value between 0-128 (0x000-0x080)
%03x	AB Mix angle	0 = mixer at A, 512 (or 0x200) = mixer at B.
%06x	V-Fade colour	RGB value

For Imagestore 300, Imagestore HD-TV and Imagestore 750 the RGB value range is 0x00 to 0x64. For Imagestore 2 the RGB range is 0x00 to 0xFF.

Set AB Cut Mode **Ub**

This command determines whether subsequent AB cuts/fades will be actioned on field 1, field 2, or any field. The default is for cuts to occur on field 1 and for fades to occur on any field.

CMD	Param 1
Ub	%02x: Mode

The following cut/fade modes are supported:

- 00: Any Field
- 01: First Field - (default)
- 02: Second Field

Example:

```
void SetABCutMode(int Mode);
{
    remote_send( "Ub%02x" , Mode );
}
```

Alternatively the command Ub responds with the AB cut mode that is currently set.

CMD
Ub

Example:

```
void EnquireABCutMode(void)
{
    remote_send( "Ub" );
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	Ub
%02x	Mode	Cut/Fade Mode

Note: When zero length AB fades are used, the timing of these events may not synchronise with “true cuts” depending on how the cut and fade modes are currently set up. A work around for this is to modify the AB fade mode to match (temporarily) the AB cut mode.

DVE Hardware Options

The Imagestore products support a range of different DVE hardware options which are summarised in the table below.

Product	Squeezy	Ross 2D/3D DVE	2D DVE
Imagestore 2	x		
Imagestore 3	x	x	
Imagestore 300/+		x	
Imagestore HD-TV			x
Imagestore 750			x (2)
LGK-3901			
DSK-3901			

Automation commands for the ‘Squeezy’ option are listed on page 115

Automation commands for the ‘Ross 2D/3D DVE’ and ‘2D DVE’ options are listed on page 119

Squeezy Commands

These commands relate to the ‘Squeezy’ hardware option for Imagestore 2 and Imagestore 3. Please refer to the table on page 113.

The Squeezy DVE is controlled by using preset positions and sizes, and performing moves to specified preset. Up to 255 presets are available (0x1 to 0xFF). Preset 0 cannot be overwritten, and is always the nominal ‘unsqueezed’ image.

Note: The Squeezy control comments only apply to Imagestore 2, and also Imagestore 3 operated with Squeezy hardware. Please refer to page 119 for all other products.

Set Squeeze Preset W0

This command modifies the basic size and position parameters for a squeeze preset. XY position and XY size are floating point values (converted to 16.16), where the active video area is defined as 1.0 high by 1.0 wide. The ‘position’ is the position of the centre of the squeezed image.

Example:

To place the image in its nominal size and position, X position = Y Position = 0.5, X Size = Y Size = 1.0.

Example:

To shrink the image to occupy the upper-right quarter of the screen:

X-Position = 0.75, Y-Position = 0.25, X-Size = 0.5, Y-Size = 0.5

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
W0	%02x : Preset No.	%08x : X Position	%08x: Y Position	%08x: X Size	%08x:Y Size

Example:

```

long oxfixed(float f)
{
    return f * 0x10000;
}

void SetPreset(int preset, float xpos, float ypos,
                float xsize, float ysize)
{
    remote_send("W0%02x%08x%08x%08x%08x", preset,
                oxfixed(xpos), oxfixed(ypos),
                oxfixed(xsize), oxfixed(ysize));
}

```

Select Squeeze Configuration W1

This determines where in the signal flow the Squeezy hardware is configured.

Note: Changing the squeeze configuration can cause instant changes in program output as it also affects keyer priorities.

CMD	Param_1
W1	%02x: Configuration Mode

Mode	Configuration Mode
0	No Squeeze
1	Picture-in-Picture (only when AB Mix option fitted – Squeeze A over B)
2	Squeeze & Reveal (swap-preview only)
3	Squeeze Stored (swap-preview only)
4	Squeeze & Reveal Midground (cascade only)
5	Squeeze & Reveal Foreground (cascade only)
6	Squeeze Midground Store (cascade only)
7	Squeeze Foreground Store (cascade only)
8	Picture-in-Picture (only when AB Mix option fitted – Squeeze B over A)
9	Squeeze Bypass (as squeeze none, but without the frame delay)

See the *Squeeze* section of the Imagestore manual for further information about squeeze modes.

Run Squeeze Move W2

Runs a squeeze move to a particular preset position over a specified duration.

CMD	Param_1	Param_2
W2	%02x : Sequence No.	%03x : Duration

Sequence number:

Sequence Number	Description
0x00	Play the currently loaded sequence in reverse
0x0x-0x61 (IS300/ISHD)	Run the specified sequence number, 1 to 99 otherwise)
0x0x-0x63	

Duration:

When not specified or set as 0x001, the sequence will play out over the default duration specified within the DVE Editor application.

When set between 0x002 and 0x3e6 (998 decimal) then the whole sequence will be stretched to play within the duration specified.

If the duration is set as 0x3e7 (999 decimal) then the sequence will be loaded without playing, and the first keyframe of the new sequence will be shown. If the sequence number is 0x00 (i.e. play in reverse) then the last keyframe of the currently loaded sequence will be shown instead.

For the Imagestore 2U/3, for which the Squeezy commands apply, the duration is measured in frames for both interlaced and progressive standards.

Select Move Profile W3

Select linear or smooth velocity profile for squeeze moves.

CMD	Param_1
W3	%01x: Profile : 0 = Linear, 1 = Smoothed

Select Preset Borders W4

Setup the border positions associated with a squeeze preset. Borders can be used to simulate wipes. For example, squeezing from full screen to a preset with a Left Border of 1.0 will ‘wipe’ the video off to the right.

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
W4	%02x : Preset No.	%08x : Left Border	%08x: Right Border	%08x: Top Border	%08x: Bottom Border

Enquire Preset W5

Returns the size, position and border information for the requested preset. Number formats are as per the W0 and W4 commands which are used to set these values

CMD	Param_1
W5	%02x : Preset No.

The information returned is:

Format	Field	Description
%c%c	Cmd	W5
%02x	Preset No.	
%08x	X Position	
%08x	Y Position	
%08x	X Size	
%08x	Y Size	
%08x	Left Border	
%08x	Right Border	
%08x	Top Border	
%08x	Bottom Border	

2D/3D DVE Commands

These commands relate to the ‘Ross 2D/3D DVE’ and ‘2D DVE’ hardware options for Imagestore 3, Imagestore 300[+], Imagestore HD-TV and Imagestore 750. Please refer to the table on page 113.

The 2D/3D DVE preset represents a time-lined sequence of user defined keyframes. We refer to this as a DVE sequence. Each keyframe contains all the information for that particular frame, for example size, position, rotation (3D-only), border softness, colour, lighting and transparency.

The 2D/3D DVE interpolates (morphs) between successive keyframes over a user specified duration. There is a limit to a total of 25 keyframes which are shared between both channels of DVE video.

All products which support for DVE allow two DVE windows to be visible on the program channel. The Imagestore 750 has two additional independent DVE windows for the preview channel.

Note: The notable functional difference between products is the DVE modes that can be selected via the W1 command (see page 120).

Backwards Compatibility Issues with Squeezy

The old Squeezy DVE supports W0, W1, W2, W3, and W4 automation commands. The 2D/3D DVE supports just the W1 and W2 automation commands from this list, plus other commands that do not apply to Squeezy (W7, W8 and WP).

This means control of the 2D/3D DVE is only partially backwards compatible with the old Squeezy. The ability to configure it is not. The configuration is instead done via the 2D/3D-DVE Editor application using extended DVE commands.

Explanation of how they work

The old Squeezy has squeeze on/ squeeze off. The term preset defines a position. The automation can set the size/position of individual presets. It can recall a preset and will take the current video size/position, and morph it to the size/position of the preset recalled.

2D/3D DVE has no squeeze on/squeeze off, it is controlled very much like a VTR its either playing, stopped, or playing in reverse. The term preset is not valid because they simply do not exist. Instead it runs a sequence which

consists of time-lined progression of keyframes which define the position/size of the video, and how to move from one keyframe to the next.

Before implementing a 2D/3D DVE automation driver, please read the section on page 125. This will explain the best way to drive the device.

Select 2D/3D DVE Configuration W1

Select the DVE configuration mode. This determines where in the program channel signal flow the 2D/3D DVE hardware is configured.

CMD	Param_1
W1	%02x: Configuration Mode

The DVE configuration modes available on the Imagestore 3 and Imagestore 300[+] 2D/3D DVE are as follows:

Mode	Description	Video Inputs (Channel 1 & 2)	Video Output Position
0x0	No Squeeze (1 Frame audio/video delay).	-	-
0x4	Squeeze & Reveal Midground. Video placed in between Midground and Foreground.	A & B	Between Midground/Foreground
0x9	Squeeze Bypass	-	-
0xA	Squeeze & Reveal Midground Mix. Video placed in between Midground and Foreground.	AB mixer & B	Between Midground/Foreground
0xB	DVE is the Midground layer and can be faded on/off like a logo	A & B	It is the Midground
0xC	DVE is the Midground layer and can be faded on/off like a logo.	B & A	It is the Midground
0xD	DVE is the Midground layer and can be faded on/off like a logo.	A & Midground fill	It is the Midground
0xE	DVE is the Midground layer and can be faded on/off like a logo.	Midground fill & A	It is the Midground
0xF	DVE is behind all layers	A & B	Behind Midground/Foreground

The DVE configuration modes available on the Imagestore HD-TV 2D DVE are as follows:

Mode	Description	Video Inputs (channel 1&2)	Video output position
0x0	None (delay) - 1 frame audio/video delay	-	-
0x1	AB Mixer	A & B	Behind both keying layers
0x4	AB over DSK1	A & B	Between DSK1 and DSK2
0x9	Bypass	-	-

The DVE configuration modes available on the Imagestore 750 2D DVE are as follows...

Mode	Description	Video Inputs (channel 1&2)	Video output position
0x0	None (delay) - 1 frame audio/video delay	-	-
0x4	Enabled	A & B	K1K2 DVE K3K4 K1 DVE K2K3K4 DVE K1K2K3K4
0x9	Bypass	-	-

When enabled, the video output position (between the available keyers) depends upon the current DVE Configuration

Note: Changing the DVE configuration can cause instant changes in program output as it also affects keyer priorities.

Load and Run Sequence W2

Loads a sequence and runs it over a specified duration. Setting the duration to 1, runs the sequence at the speed it was designed. The maximum duration is 999 (0x3E7). The 2D/3D DVE holds up to 97 (0x61) sequences. Running sequence 0 plays the currently loaded sequence backwards. Some 2D/3D warp sequences require a 2 frame setup time, so it is recommended loading the first keyframe (999 (0x3E7)), then switching the DVE in circuit and then running the sequence.

CMD	Param_1	Param_2 (Optional)	Param_3 (Optional)
W2	%02x: Sequence No.	%03x: Duration 0x001=Default 0x3E7=First frame	%1x: DVE Mask (Imagestore 750 only) 0x1=Program, 0x2=Preview, 0x3=Program and preview

Sequence number:

Sequence Number	Description
0x00	Play the currently loaded sequence in reverse
0x01x-0x61	Load and run the specified sequence number
0x01x-0x63 (IS3 only)	

Sequence 0 is always used to play the current sequence in reverse. So if you have a sequence which takes full screen video and squeezes it down to a small box, then re-calling sequence 0 will take it back to full screen video.

Duration:

The product and standard determine the meaning of the duration parameter as follows:

Product	Standard	
	Interlaced	Progressive
Imagestore 2U/3	Frames	n/a
Imagestore 300	Frames	n/a
Imagestore HD-TV	Fields	Frames
Imagestore 750	Fields	Frames

DVE Mask:

For Imagestore 750 there is an optional bitwise mask that allows automation to define whether commands apply to the program and/or preview channels. For all other products and for Imagestore 750 when this parameter is not used the DVE commands only apply to the program channel.

Run Sequence W7

Runs the currently loaded sequence over a specified duration. Setting the duration to 1, runs the sequence at the speed it was designed at within the DVE Editor application. The maximum duration is 999 (0x3E7). The duration depends on the product and standard, as defined within the W2 section. The DVE mask option is available for Imagestore 750 only, and is as defined within the W2 section.

CMD	Param_1	Param_2 (Optional)
W7	%03x : Duration 1 = Default	%1x: DVE Mask (Imagestore 750 only) 0x1=Program, 0x2=Preview, 0x3=Program and preview

Run Sequence in Reverse W8

Runs the currently loaded sequence in reverse over a specified duration.

Setting the number of frames/fields to 1, runs the sequence at the speed it was designed at. Maximum duration is 999 (0x3E7). The duration depends on the product and standard, as defined within the W2 section. The DVE mask option is available for Imagestore 750 only, and is as defined within the W2 section.

CMD	Param_1	Param_2 (Optional)
W8	%03x : Duration 1 = Default	%1x: DVE Mask (Imagestore 750 only) 0x1=Program, 0x2=Preview, 0x3=Program and Preview

Select 2D/3D DVE Preview Configuration WP

This command only applies to Imagestore 750 and sets a DVE configuration for the Preview channel – (it is equivalent to the W1 command which only applies to the Program channel)

Note: Changing the DVE configuration can cause instant changes in program output as it also affects keyer priorities.

CMD	Param_1
WP	%02x: Configuration Mode

The modes available are as follows:

Mode	Description	Preview Video Inputs (channel 1&2)	Preview Video output position
0x0	None (delay) - 1 frame audio/video delay	-	-
0x4	Enabled	A & B	K1K2 DVE K3K4 K1 DVE K2K3K4 DVE K1K2K3K4
0x9	Bypass	-	-

Select DVE Routing Input WR

This command only applies to Imagestore 750 and sets the input the to program DVE or the preview DVE. Note that the number of DVE inputs and sources supported depends on licences:

CMD	Param_1	Param_2	Param_3
WR	%1x: DVE Mask 1 – Program DVE 2 – Preview DVE	%1x: DVE Input 0 – DVE 1 input 1 – DVE 2 input	%1x: Source 0 – A input 1 – B input 2 – AB mix input 3 – Fill 1 input 4 – Key 1 input 5 – Fill 2 input 6 – Key 2 input 7 – C input 8 – D input

The Imagestore 750 DVE routing can be enquired:

CMD	Param_1	Param_2
WR	%1x: DVE Mask 1 – Program DVE 2 – Preview DVE	%1x: DVE Input 0 – DVE 1 input 1 – DVE 2 input

The information returned by the Imagestore 750 is as follows, where the parameter options are as defined above:

Format	Field	Description
%c%c	Cmd	WR
%1x	DVE Mask	Program DVE or Preview DVE
%1x	DVE Input	DVE input
%1x	Source	SDI input source

How to use 2D/3D DVE via Automation

In normal Imagestore operation the 2D/3D DVE is not in circuit and the DVE mode is set to none (0). DVE none adds a one frame of video/audio delay so that when the DVE is switched in and out of circuit there is no jump in the video.

Before switching the 2D/3D DVE in circuit, it is often required to start on full screen A. The best way to do this is to make sure all sequences start on full screen A. The automation command W2 (load and run) with certain parameters can be used to load the first key frame of a sequence without running it. This has affect of queuing it up ready to run.

Some warp sequences have a two frame setup delay, so the use of queuing up a move is essential when using warp moves. When the move is queued up, you can then switch the DVE in circuit, and run the sequence.

Example use:

W100: Set 2D/3D DVE to mode none (out of circuit but with delay).

W2603E7: Load the first key frame of sequence number 96.

W104: Switch 2D/3D DVE in circuit.

W7001: Run the sequence currently loaded at the speed it was designed at. (Note that use of W7 is recommended instead of W2).

Wait for DVE sequence to finish.

W100 Set 2D/3D DVE to mode none when the sequence has finished.

Easyplay Commands

This set of commands is only available if the unit has the Easysound audio mixer and Easyplay options installed.

Load Audio Clip k0

Loads an audio clip in preparation for playout.

CMD	Param_0	Param_2
k0	%01x Layer No.	%s Audio file name

The following layers are supported:

0: Audio store 1, associated with DSK1

1: Audio store 2, associated with DSK2

For Imagestore 750 a further two layers are supported:

2: Audio store 3, associated with DSK 3

3: Audio store 4, associated with DSK 4

Please see page 269 for per-product layer descriptions

Example:

```
void load_audio(char * filename, int layer)
{
    char temp[100];
    put_remote_str(filename, temp);
    remote_send("k0%1x%s", layer, temp);
}
```

Start Playout k1

Start playout of an audio clip

CMD	Param_1
k1	%1x Layer No.

Please note that only one Easysound layer can play at a time since a single voice-over is shared between layers.

Stop Playout k2

Stop playout of an audio clip

CMD	Param_1
k2	%1x Layer No.

Unload Audio Clip k3

Unloads an audio clip from its store slot. This prevents the audio clip from being played out automatically when its associated keyer is cut up.

CMD	Param_1
k3	%1x Layer No.

Enquire Audio Clip k4

Enquires the loaded Easyplay audio clip name for a given store.

CMD	Param_1
k4	%1x Store No.

Extended Status Commands

Temperature Monitoring X0

Returns information about the current temperature within the unit.

CMD
X0

The information returned is:

Format	Field	Description
%c%c	Cmd	X0
%5f	Temperature	In degrees Celsius

Enquire Licence Code X1

Note: LGK-3901 and DSK-3901 licences cannot be enquired using the X1 command, and will return "X100000" to all enquiries. Use iControl to view the licences.

This command allows remote systems to inquire whether a unit has a particular licence installed. This can be used to detect if units are fitted with options such as Easysound, Squeezy and AB/Mix front-end..

CMD	Param_1
X1	%c%c%c%c: - Licence code to query

Example:

```
void EnquireEthernet()
{
    remote_send( "X1ENET" );
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	X1
%c%c%c%c	Licence	Confirmation of code.
%01x	Valid	0 = Licence invalid, 1 = Licence valid
%04x	Licence Value	Value associated with the licence

Insert Log Message X2

Adds a text message to the Imagestore unit's log file. This log can be interrogated and viewed by support engineers, and recent log entries can be seen on the Imagestore's preview output when in Swap/Preview mode.

CMD	Param_1
X2	%s: Message

Example:

```
void InformStartup()
{
    remote_send("X2%s", "Automation System
rebooted");
}
```

Enquire Command Availability X3

This command can be used to determine if a particular command is supported and is available to use by a particular machine. For example if a hardware option e.g. DVE fails to initialise on start up, the DVE commands will not be available.

CMD	Param_1
X3	%2c : Command byte(s)

The information returned is:

Format	Field	Description
%c%c	Cmd	X3
%c%c	Cmd	Confirmation of command code.
%01x	Supported	0 = Unsupported, 1=Supported

Unit Bypass X4

This command allows automation control of the unit's optional mechanical video relay. If Flag is 1, the unit is bypassed, if 0, the unit is in circuit.

CMD	Param_1
X4	%1x: Flag, 0=Bypassed, 1=In circuit

Return Input Colour Field Values X5

This command returns information about the various input colour field values on an Imagestore.

CMD
X5

The information returned on Imagestore 300 and Imagestore HD-TV is:

Format	Field	Description
%c%c	Cmd	X5
%01x	A mode	1 = Colour field, 0 = Other
%06x	A colour field	RGB value
%01x	Fill 1 mode	1 = Colour field, 0 = Other
%06x	Fill 1 colour field	RGB value
%01x	Key 1 mode	1 = Colour field, 0 = Other
%06x	Key 1 colour field	RGB value
%01x	B mode	1 = Colour field, 0 = Other
%06x	B colour field	RGB value
%01x	Fill 2 mode	1 = Colour field, 0 = Other
%06x	Fill 2 colour field	RGB value
%01x	Key 2 mode	1 = Colour field, 0 = Other
%06x	Key 2 colour field	RGB value

If there are no Fill 2/Key 2 inputs, then they are treated as colour-field disabled and the colour is returned as black.

For Imagestore 750 the command is extended to include extra SDI inputs (Fill-3 and Key-3). Also there are only three colour fields (shared, not dedicated) each of which may be associated with any physical input(s). The RGB information returned applies to the colour field currently associated with the SDI input (if any).

Format	Field	Description
%c%c	Cmd	X5
%01x	A mode	1 = Colour field, 0 = Other
%06x	A colour field	RGB value
%01x	Fill 1 mode	1 = Colour field, 0 = Other
%06x	Fill 1 colour field	RGB value
%01x	Key 1 mode	1 = Colour field, 0 = Other
%06x	Key 1 colour field	RGB value
%01x	B mode	1 = Colour field, 0 = Other
%06x	B colour field	RGB value
%01x	Fill 2 mode	1 = Colour field, 0 = Other

%06x	Fill 2 colour field	RGB value
%01x	Key 2 mode	1 = Colour field, 0 = Other
%06x	Key 2 colour field	RGB value
%01x	C/Fill 3 mode	1 = Colour field, 0 = Other
%06x	C/Fill 3 colour field	RGB value
%01x	D/Key 3 mode	1 = Colour field, 0 = Other
%06x	D/Key 3 colour field	RGB value

Enquire Voltages **X6**

Note: LGK-3901 and DSK-3901 do not support the X6 command.

This command returns information (separated by spaces) about the various voltages on an Imagestore.

CMD
X6

The information returned on Imagestore 2, Imagestore 300 is:

Format	Field	Description
%c%c	Cmd	X6
%05.1f	Temperature	In degrees Celsius
%03.2f	-12V supply	
%03.2f	-5V supply	
%03.2f	+3.3V supply	
%03.2f	+5V supply	
%03.2f	+5V	Analogue supply
%03.2f	+12V	
%03.2f	PLL Voltage	

The information returned on Imagestore HD-TV is:

Format	Field	Description
%c%c	Cmd	X6
%05.1f	Temperature	In degrees Celsius
%03.2f	-12V supply	
%03.2f	+1.8V supply	
%03.2f	+3.3V supply	
%03.2f	+5V supply	
%03.2f	+2.5V	
%03.2f	+12V	
%03.2f	0.00	Unused field

The information returned on Imagestore 750 is:

Format	Field	Description
%c%c	Cmd	X6
%05.1f	Ambient Temperature	In degrees Celsius
%03.2f	12V PSU A supply	1.00=OK, 0.00=Failed or not fitted
%03.2f	12V PSU B supply	1.00=OK, 0.00=Failed or not fitted
%03.2f	12V supply	In volts. Worst case measurement
%03.2f	Video Voltage Check	1.00=OK, 0.00=Failed. Checks all video board voltages are within tolerances
%03.2f	DVE Voltage Check	1.00=OK, 0.00=Failed. Checks all DVE board voltages are within tolerances
%03.2f	0.00	Unused field
%03.2f	Fan Speed	In RPM. Worst case fan 1-5 status

Voltage tolerances are +/-10% for the 12V supplies and +/-5% for all other voltages.

Virtual GPI X7

Triggers the GPI macro associated with the GPI. It is called a virtual GPI because it does not affect the physical GPI in any way. If it is required to set the physical GPI use the X9 command.

CMD	Param_1	Param_2
X7	%01x On/Off	%02x GPI Number

GPI Input Status X8

Note: For Imagestore 300[+] or Imagestore HD-TV units with GPI lines on video and audio cards, the video GPIs are numbered 0 to 8 and the audio GPIs are numbered 9 to 17. For the Imagestore IS750 only video GPIs are supported, numbered 0 to 15. For the LGK-3901 and DSK-3901 only video GPIs 0 to 7 are supported. Please check user manuals for details.

This command will respond with the status of a GPI line. The GPI number is the zero based GPI line number between 0 and 7

CMD	Param_1	Param_2
X8	%01x 0 = Video GPI Lines 1 = Audio GPI Lines (if supported)	%02x: GPI Number

Response format

CMD	Param_1	Param_2
X8	%02x: GPI No.	%01x: 0=GPI off (level high), 1=GPI on (level low)

Example:

```
INT rem_remote_gpi_read(Premote_ rem, INT8 cmd, char * str)
{
    BOOL video_gpi[MAX_GPI], audio_gpi[MAX_GPI];
    int result = -1, vid_aud = -1, number = -1;

    if (sscanf(str, "%01x%02x", &vid_aud, &number) == 2)
    {
        if (number < MAX_GPI)
        {
            get_gpi_state(video_gpi, audio_gpi);

            if (vid_aud == 0)
            {
                result = video_gpi[number];
            }
            else
            {
                result = (audio_fitted ? audio_gpi[number] : -1);
            }
        }

        if (result != -1)
        {
            rem_send_status(rem, "X8%02x%01x", number, result);
        }
        else
        {
            // 99 indicates something has gone wrong
            rem_send_status(rem, "X8%02x%01x", 99, 99);
        }
    }
}
```

GPI Special X9

This command can perform two functions. One of the functions is to return a status message containing a bit mask of the GPI input status. The other function is to set the physical output status of a GPI.

CMD	Param_1	Param_2
X9	%01x 0=Turn GPI off 1=Turn GPI on 2=Request Status	%02x GPI number (Not required for status request).

Note: For Imagestore 300[+] or Imagestore HD-TV units with GPI lines on video and audio cards, the video GPIs are numbered 0 to 8 and the audio GPIs are numbered 9 to 17. For the Imagestore IS750 only video GPIs are supported, numbered 0 to 15. For the LGK-3901 and DSK-3901 only video GPIs 0 to 7 are supported. Please check user manuals for details.

If a status request is sent the information returned is

Format	Field	Description
	Cmd	X9
%06x	GPI Status	Current GPI status

Enquire Extended Intuition Layer status XA

This command returns a status block about the layer status of the Imagestore and Intuition. As the Imagestore can only control the first 14 layers of the Intuition only the status of those layers are returned. The layer status of the first two Imagestore layers is also included in the response for completeness. Note that the Intuition layer status is based purely on the commands the Imagestore has forwarded to the Intuition. This means if the Intuition is being controlled directly either by automation or the front panel, the changes will not be reflected in this command.

CMD
XA

Example:

```
void EnquireExtLayerStatus(void)
{
    remote_send( "XA" );
}
```

The information returned is:

Format	Field	Description
	Cmd	XA
%c	Layer (0=DOWN, 1=UP)	Imagestore Midground Layer (DSK 0)
%c	Layer	Imagestore Foreground Layer (DSK 1)
%c	Layer	Intuition Layer 0 (0x0)
%c	Layer	Intuition Layer 1 (0x1)
%c	Layer	Intuition Layer 2 (0x2)
%c	Layer	Intuition Layer 3 (0x3)
%c	Layer	Intuition Layer 4 (0x4)
%c	Layer	Intuition Layer 5 (0x5)
%c	Layer	Intuition Layer 6 (0x6)
%c	Layer	Intuition Layer 7 (0x7)
%c	Layer	Intuition Layer 8 (0x8)
%c	Layer	Intuition Layer 9 (0x9)
%c	Layer	Intuition Layer 10 (0xA)
%c	Layer	Intuition Layer 11 (0xB)
%c	Layer	Intuition Layer 12 (0xC)
%c	Layer	Intuition Layer 13 (0xD)

Example of returned XA is: “XA0110000000000000”

Representing, the DSK 1 of the Imagestore is cut up, and the first layer of the intuition is cut up.

GPI Output Status XB

This command sets the output status of a particular GPI number. The GPI number range is 1-7, or 1-14 when an audio card is fitted in the Imagestore 300[+] or Imagestore HD-TV. The GPI number range is 1-16 in the Imagestore 750, where the XB commands are zero based (0-15) for this product. The GPI number range and XB command is 0-7 for the LGK-3901 and DSK-390.

CMD	Param_1	Param_2
XB	%01x: GPI No.	%01x: 0 = Off, 1 = On

Example:

```
void GPIOOutputStatus(int GPI, bool Status)
{
    remote_send("XB%01x %01x", GPI, Status);
}
```

Serial Baud Rate XC

This command sets the baud rate for the specified serial port number.

CMD	Param_1	Param_2
XC	%1x: Port No.	%d: Baud rate

Example:

```
void SetSerialBaudRate(int Port, int Baudrate)
{
    remote_send("XC%1x %d", Port, Baudrate);
}
```

The command XC alternatively responds with the current baud rate for the serial port specified.

CMD	Param_1
XC	%1x: Port No.

Example:

```
void GetSerialBaudRate(int Port)
{
    remote_send("XC%1x", Port);
}
```

Format	Field	Description
%c%c	Cmd	XC
%x	Port	Serial Port No.
%d	Baud rate	Current baud rate

Time Code Source XE

This command sets the current Time Code source choosing between two possible values, 0 = Linear Time Code (LTC) and 1 = Vertical Interrupt Time Code (VITC).

CMD	Param_1
XE	%x: Source

Example:

```
void SetTimeCodeSource(int Source)
{
    remote_send("XE%x", Source);
}
```

The command XE alternatively responds with information about the time code source.

CMD
XE

For Imagestore 300[+], Imagestore HD-TV or Imagestore 750 the information returned is

Format	Field	Description
%c%c	Cmd	XE
%x	Source	Time code source 0 = Linear Time Code (LTC) 1 = Vertical Interrupt Time Code (VITC).

For LGK-3901 or DSK-3901 the information returned is

Format	Field	Description
%c%c	Cmd	XE
%x	Source	Time code source 0 - Linear Time Code (LTC) 1 - Vertical Interrupt Time Code (VITC).
%d	Valid	Time code status 0 – Invalid 1 – Valid

Set Preview XH

This command sets the current preview mode as follow:

CMD	Param_1
XH	%02x: Preview mode

The supported preview modes are:

- 0x00: DSK 1 Key o/p
- 0x01: DSK 2 Key o/p
- 0x02: DSK 1 o/p
- 0x03: Program o/p
- 0x04: DSK 2 Fill o/p
- 0x05: DSK 1 Fill o/p
- 0x06: SDI A
- 0x07: Fill-1
- 0x08: Key-1
- 0x09: SDI B
- 0x0A: AB Mixer o/p
- 0x0B: Presmaster Preview
- 0x0C: Fill-2 (Optional)
- 0x0D: Key-2 (Optional)

For Imagestore 750 the supported preview modes are:

- 0x02: PGM DSK 1 output
- 0x03: Program output
- 0x0A: Program AB Mixer output
- 0x0B: Presmaster Preview
- 0x0E: PGM DVE output
- 0x0F: PGM DSK 2 output
- 0x10: PGM DSK 3 output

For LGK-3901 the supported preview modes are:

- 0x03: Program output
- 0x0B: Preview output
- 0x02: PGM DSK 1 output
- 0x0F: PGM DSK 2 output
- 0x10: PGM DSK 3 output
- 0x11: PGM DSK 4 output
- 0x07: SDI Fill-1
- 0x08: SDI Key-1
- 0x0C: SDI Fill-2
- 0x0D: SDI Key-2
- 0x05: Store 1 Fill

- 0x00: Store 1 Key
- 0x04: Store 2 Fill
- 0x01: Store 2 Key
- 0x13: Store 3 Fill
- 0x14: Store 3 Key

For DSK-3901 the supported preview modes are:

- 0x03: Program output
- 0x0B: Preview output
- 0x02: PGM DSK 1 output
- 0x07: SDI Fill-1
- 0x08: SDI Key-1
- 0x0C: SDI Fill-2
- 0x0D: SDI Key-2

Example:

```
void SetPreview(int Preview)
{
    remote_send("XH%02x", Preview);
}
```

The command XH alternatively responds with the current preview mode.

Format	Field	Description
%c%c	Cmd	XH
%02x	Preview	Current preview mode

Set Reference XI

This command sets the current video signal reference mode as follows:

- 0: Analogue (REF1)
- 1: Analogue (REF2)
- 2: SDI-BGnd
- 3: Video genlock is current standard

Note: For Intuition[+], the XI command can not be used to set the video signal reference since this is automatically detected by the video card.

For Imagestore 750, LGK-3901 and DSK-3901 the only external reference supported is "0". For these products the reference is automatically selected and cannot be set.

CMD	Param_1
XI	%1x: Reference

Example:

```
void SetReference(int Reference)
{
    remote_send("XI%x", Reference);
}
```

The command XI alternatively responds with the current reference mode.

Format	Field	Description
%c%c	Cmd	XI
%x	Reference	Current reference mode

For Intuition[+], the XI enquire command will return one of the following:

- 3: Video genlock is current standard
- 4: Invalid reference signal
- 5: Video genlock is valid but not the current standard

Clock Offset XJ

This command sets the current clock offset for the specified reference mode as follows:

- 0: Analogue (REF1)
- 1: Analogue (REF2)
- 2: SDI-BGnd
- 3: Video genlock is current standard

CMD	Param_1	Param_2
XJ	%1x: Reference	%06x: Clock Offset

Note: For Imagestore 750, LGK-3901 and DSK-3901 the only external reference supported is “0”. For these products the reference is automatically selected and cannot be set.

Example:

```
void SetClockOffset(int Reference,
                     int ClockOffset)
{
    remote_send("XJ%1x%06x", Reference,
                ClockOffset);
}
```

The command XJ alternatively responds with the current clock offset related to the reference mode.

Format	Field	Description
%c%c	Cmd	XJ
%x	Reference	Current reference mode
%06x	Clock Offset	Current clock offset for the reference
%06x	Minimum	Minimum accepted reference value (-1 indicates no minimum)
%06x	Maximum	Maximum accepted reference value (-1 indicates no maximum)

Get Serial Number XK

This command returns the eight digit serial number of a unit.

CMD
XK

Example:

```
void GetSerialNumber()
{
    remote_send("XK");
}
```

Then the command will return the serial number as follow:

Format	Field	Description
%c%c	Cmd	XK
%08d	Serial Number	Imagestore Serial Number

The highest four digits of the serial number define the product code, and the lowest four digits define the unit ID. The exception is Imagestore 750 where the four digit product code is followed by a zero then a three digit unit ID. The full serial number for an Imagestore 750 can be obtained using the XU command (see page 149).

Product codes are as follows:

- 0629: Imagestore 2
- 0675: Imagestore HD-TV
- 0730: Imagestore 3
- 0765: Imagestore 300
- 0781: Intuition SD
- 0811: Intuition HD
- 0814: Imagestore 300+ 1RU
- 0832: Imagestore 300+ 2RU
- 0838: Intuition+ SD
- 0839: Intuition+ HD
- 0872: Imagestore 750
- 0887: LGK-3901, DSK-3901

System Halt XL

This command requests a system halt as soon as the OS can process the request.

CMD
XL

Example:

```
void SystemHalt( )
{
    remote_send( "XL" );
}
```

Input Mode XM

This command configures an input to pass SDI or a colour field and takes the following format where the input parameter is system dependent as per the table on page 47.

CMD	Param_1	Param_2
------------	----------------	----------------

XM	%1d: Input	%1d: Mode
-----------	-------------------	------------------

The command behaviour depends on the model of Imagestore, as follows:

Imagestore 300[+] and Imagestore HD-TV

Each input has a dedicated colour field set via the L command (page 48). The supported modes are therefore:

- 0: Pass SDI
- 1: Force Colour Field

Imagestore 750

There are three independent colour fields (again set by the L command) that can be routed to any input(s). In addition two built-in colour fields may be routed to any input(s). The supported modes for each input are therefore:

- 0: Pass SDI
- 1: Force Colour Field 1
- 2: Force Colour Field 2
- 3: Force Colour Field 3
- 4: Force Test Pattern 1
- 5: Force Test Pattern 2

Example:

```
void SetInputMode(int Input, int Mode)
{
    remote_send("XM%1d %1d", Input, Mode);
}
```

The command XM alternatively responds with the mode associated with the desired input.

Format	Field	Description
%c%c	Cmd	XM
%d	Input	Input
%d	Mode	Input mode associated

Serial Protocol XN

This command sets the protocol for the specified serial port number. It is closely related to the XC command which sets the baud rate of a given serial port.

CMD	Param_1	Param_2
XN	%1x: Port number	%1x: Protocol

The following protocols are supported:

- 0: Automation (server to any client)
- 1: Presmaster (server to Presmaster client)
- 2: Easysound (client of Easysound Standalone server)
- 3: Intuition (client of Intuition clip-on server)
- 4: Sage ENDEC Model 1822
- 5: TFT EAS 911T

The following protocols are supported by LGK-3901 and DSK-3901:

- 0: Automation (server to any client)
- 3: Intuition (client of Intuition clip-on server)
- 4: Sage ENDEC Model 1822
- 5: TFT EAS 911T

Example:

```
void SetSerialProtocol(int Port, int Protocol)
{
    remote_send("XN%1x%1x", Port, Protocol);
}
```

The command XN alternatively responds with the current protocol for the serial port specified.

CMD	Param_1
XN	%1x: Port number

Example:

```
void GetSerialProtocol(int Port)
{
    remote_send("XN%1x", Port);
}
```

Format	Field	Description
%c%c	Cmd	XN
%x	Port	Serial Port number
%x	Protocol	Current Protocol

Clean Feed **XP**

This command sets one of the auxiliary channels as the current clean feed input, if supported.

CMD	Param_1
XP	%2x: Clean Feed

Possible settings for the Imagestore 300+ clean feed channel are as follows:

- 0x00: Program (factory default)
- 0x01: SDI A
- 0x02: SDI B
- 0x03: AB Mixer
- 0x04: AB Mixer + DVE
- 0x05: DSK 1

Possible settings for the Imagestore 750 clean feed channel are as follows:

- 0x00: Program (factory default)
- 0x01: SDI A
- 0x02: SDI B
- 0x03: SDI C
- 0x04: SDI D
- 0x05: Program AB Mixer
- 0x06: Program DVE output
- 0x07: Program DSK 1 output
- 0x08: Program DSK 2 output
- 0x09: Program DSK 3 output

Please note that for IS750 LOGO this command will be ineffective unless the ISCF licence is enabled

Example:

```
void SetCleanFeed(int CleanFeed)
{
    remote_send("XP%2x", CleanFeed);
}
```

The command XP alternatively responds with the current clean feed channel.

Format	Field	Description
%c%c	Cmd	XP
%2x	Clean Feed	Current clean feed channel

Get Total Fill/Key XR

This command returns how many Fill & Key are currently supported by the Imagestore. The returned value is normalised from zero.

CMD
XR

Format	Field	Description
%c%c	Cmd	XR
%2x	F+K	Total number of Fill & Keys supported

Get Total Preview XS

This command returns how many previews are currently supported by the Imagestore.

CMD
XS

Format	Field	Description
%c%c	Cmd	XR
%2x	Preview	Total number of previews supported

Monitor Feed XT

This command sets the feed displayed by the monitor for an Imagestore 750:

CMD	Param_1
XT	%02x: Monitor Feed

Possible settings for the Imagestore 750 monitor feed are as follows:

- 0x00: Program (factory default)
- 0x01: Preview
- 0x02: SDI A
- 0x03: SDI B
- 0x04: SDI C
- 0x05: SDI D
- 0x06: Program AB Mixer
- 0x07: Preview AB Mixer
- 0x08: Store 1 Fill
- 0x09: Store 1 Key
- 0x0A: Store 2 Fill
- 0x0B: Store 2 Key
- 0x0C: Store 3 Fill
- 0x0D: Store 3 Key
- 0x0E: Store 4 Fill
- 0x0F: Store 4 Key
- 0x10: SDI Fill-1
- 0x11: SDI Key-1
- 0x12: SDI Fill-2
- 0x13: SDI Key-2
- 0x14: Colour 1
- 0x15: Colour 2
- 0x16: Colour 3
- 0x17: Colour 4 (V-Fade Colour)
- 0x18: Test Pattern 1
- 0x19: Test Pattern 2
- 0x1A: Program DVE 1
- 0x1B: Program DVE 2
- 0x1C: Preview DVE 1
- 0x1D: Preview DVE 2
- 0x1E: Program DVE output
- 0x1F: Program DSK 1 output
- 0x20: Program DSK 2 output
- 0x21: Program DSK 3 output

Please note that:

- DSK fill and key outputs are routed from the stores, not the keyers
- For IS750 LOGO this command will be ineffective unless the ISMF licence is enabled.

Get Extended Serial Number XU

This command returns the full fifteen digit extended serial number of an Imagestore 750.

CMD
XU

Then the command will return the extended serial number as follows:

Format	Field	Description
%c%c	Cmd	XU
%15s	Serial Number	Imagestore extended serial number

The extended serial number will be of the form "PPPPEE-WWWWWSSS", as shown on the front panel under "System Information", where:

P: Project number (4 digits) (see command XK)

E: Final assembly element number (2 digits)

W: Works order number (5 digits)

S: Actual serial number (3 digits)

Get Extended System Health XV

This command returns the extended health status of an Imagestore 750.

CMD
XV

The values broadcast in the response are separated by spaces.

Format	Field	Description
%c%c	Cmd	XV
%05.1f	Ambient Temperature	In degrees Celsius. The worst case temperature measurement
%05.1f	CPU Temperature	In degrees Celsius
%05.1f	KID Temperature	In degrees Celsius
%05.1f	CPU FPGA Temperature	In degrees Celsius
%05.1f	SID Temperature	In degrees Celsius
%05.1f	SOD Temperature	In degrees Celsius
%05.1f	DVE1 Temperature	In degrees Celsius (when fitted)
%05.1f	DVE2 Temperature	In degrees Celsius (when fitted)
%03.2f	12V PSU A Supply	In volts
%03.2f	12V PSU B Supply	In volts
%03.2f	5V Supply	In volts
%03.2f	3.3V Supply	In volts
%03.2f	2.5V Supply	In volts
%03.2f	SID 1.8V Supply	In volts
%03.2f	1.2V Supply	In volts
%03.2f	DVE 12V Supply	In volts (when fitted)
%03.2f	DVE 2.5V Supply	In volts (when fitted)
%03.2f	DVE 1.8V Supply	In volts (when fitted)
%03.2f	DVE 1.2V Supply	In volts (when fitted)
%01x	PSU A Status	1=OK, 0=failed
%01x	PSU A Converted Status	1=OK, 0=failed
%01x	PSU B Status	1=OK, 0=failed
%01x	PSU B Converted Status	1=OK, 0=failed
%03.2f	Fan 1 speed	In RPM
%03.2f	Fan 2 speed	In RPM
%03.2f	Fan 3 speed	In RPM
%03.2f	Fan 4 speed	In RPM
%03.2f	Fan 5 speed	In RPM

The XV command is related to the enquire voltages command (X6). Presmaster and IMM actually utilise X6 for sounding alarms, but XV should then be used to find the real reasons for the alarm.

Get Input Status **XX**

This command returns the current status of the Imagestore 750, DSK-3901 and LGK-3901 input requested:

CMD	Param_1
XX	%1x: Input

Where the Imagestore 750 inputs are numbered in the following order.:

- 0: A
- 1: Fill 1
- 2: Key 1
- 3: B
- 4: Fill 2
- 5: Key 2
- 6: C
- 7: D

For the DSK-3901 and LGK-3901 inputs are numbered as follows:

- 0: Program In
- 1: Fill 1
- 2: Key 1
- 4: Fill 2
- 5: Key 2

The Imagestore 750 return uses the following format:

Format	Field	Description
%c%c	Cmd	XR
%1x	Input	Input enquired
%1x	Status	Status of input, 0=unlocked, 1=locked
%1x	Standard	Standard of input, 0 = PAL, 1 = NTSC, 2 = 1080i 59.94Hz, 3 = 1080i 50Hz, 4 = 720p 59.94Hz, 5 = 720p 50Hz, f = Unknown (e.g. no input connected)

The DSK-3901 and LGK-3901 return uses the following format:

Format	Field	Description
%c%c	Cmd	XR
%1x	Input	Input enquired
%1x	Status	Status of input, 0=unlocked, 1=locked
%1x	Standard	Standard of input, 0 = PAL, 1 = NTSC, 2 = 1080i 59.94Hz, 3 = 1080i 50Hz, 4 = 720p 59.94Hz, 5 = 720p 50Hz, 5 = 1080p 59.94Hz at level A, 6 = 1080p 50Hz at level A, 7 = 1080p 59.94Hz at level B, 8 = 1080p 50Hz at level B, f = Unknown (e.g. no input connected)

Pass Close Caption Xc

This command enables or disables the pass close caption data in the video signal.

CMD	Param_1
Xc	%1x: 1 = Enable, 0 = Disable

Example:

```
void PassCloseCaption(bool Enable)
{
    remote_send("Xc%1x", Enable);
}
```

The command Xc alternatively responds with the current pass close caption flag.

Format	Field	Description
%c%c	Cmd	Xc
%1x	Enable	Current pass close caption flag

Set Time/Date Xe

This command sets the time/date:

CMD	Param_1	Param_2	Param_3	Param_4	Param_5	Param_6
Xe	%04d: Year	%02d: Month	%02d: Day	%02d: Hour	%02d: Minute	%20d: Second

For example Xe20070302230010 Sets the date to be 2nd March 2007 and the time to be 23:00:10.

Store-Keyer Configuration Xi

Note: Changing the store-keyer configuration unloads all stores. The store-keyer configuration cannot be changed when EAS is active. This command is only supported on the LGK-3901.

This command sets the store-keyer configuration:

CMD	Param_1
Xi	%1x: Store-Keyer Configuration

The possible LGK-3901 store-keyer configurations are:

Param_1	Keyer Number					Representation
	1	2	3	4	5	
0	Store	Store	Store			1S 2S 3S 4 5
1	Store	Store		Store		1S 2S 3 4S 5
2	Store	Store			Store	1S 2S 3 4 5S
3	Store		Store	Store		1S 2 3S 4S 5
4	Store		Store		Store	1S 2 3S 4 5S
5	Store			Store	Store	1S 2 3 4S 5S
6		Store	Store	Store		1 2S 3S 4S 5
7		Store	Store		Store	1 2S 3S 4 5S
8		Store		Store	Store	1 2S 3 4S 5S
9			Store	Store	Store	1 2 3S 4S 5S

The current store-keyer configuration can be enquired using:

CMD
Xi

Response:

Format	Field	Description
%c%c	Cmd	Xi
%1x	Store-Keyer Configuration	The current store-keyer configuration, as indicated above

Set Standard Xs

This command sets the standard:

CMD	Param_1
Xs	%x Standard

The standard refers to:

0	PAL
1	NTSC
2	1080i 59.94 Hz
3	1080i 50Hz
4	720p 59.94Hz
5	720p 50Hz

For LGK-3901, DSK-3901 with SD, HD and 3G the standard refers to:

0	PAL
1	NTSC
2	1080i 59.94 Hz
3	1080i 50Hz
4	720p 59.94Hz
5	720p 50Hz
6	1080p 59.94 Level A
7	1080p 50 Level A

The standard can be enquired using:

CMD
Xs

The return is consistent with the set standard format.

Set Frame Delay Xt

This command sets the output frame delay:

CMD	Param_1	Param_2
Xt	%1x Output	%02x Frames

The output refers to:

0	Program
1	Preview
2	Clean Feed

Please note that there is no delay capability for the Monitor Feed in Imagestore 750. LGK-3901 and DSK-3901 do not support frame delays.

The frame delay can also be enquired using:

CMD	Param_1
Xt	%1x Output

The return is consistent with the set frame delay format.

Get AES Audio Signal Type Xu

Note: This command is only supported by Imagestore 750 v2.0 or higher.

This command gets the AES audio signal type for a given channel

CMD	Param_1
Xu	%02x AES Input Channel Pair

The information returned for Imagestore 750 is:

Format	Field	Description
%c%c	Cmd	Xu
%02x	AES Audio Type	Audio type 0 - PCM audio 1 - Non-PCM audio 2 - DolbyE audio 3 - DolbyD audio

Get Embedded Audio Signal Type **Xv**

Note: This command is only supported by Imagestore 750 v2.0 or higher.

This command gets the embedded audio signal type for a given channel:

CMD	Param_1	Param_1
Xv	%02x Audio Feed 0 = A, 1 = B, 2 = C, 3 = D	%02x Audio Channel No. (0 to 15)

The information returned for Imagestore 750 is:

Format	Field	Description
%c%c	Cmd	Xv
%02x	Audio Feed	Audio input feed 0 – audio feed A 1 – audio feed B 2 – audio feed C 3 – audio feed D
%02x	Channel No.	Audio channel number (0-15)
%x	Audio Type	Audio type 0 - PCM audio 1 - Non-PCM audio 2 - DolbyE audio 3 - DolbyD audio

LGK-3901 and DSK-3901 only supports PCM audio as an audio input type.

Import Configuration File **Xx**

This command allows a configuration file to be imported onto the Imagestore 750, DSK-3901 and LGK-390:

CMD	Param_1
Xx	%s: Configuration file including path name

Example:

```
void SetCleanFeed(int CleanFeed)
{
    remote_send("Xx%s", "/home/configs/Name.xml");
```

The information returned indicates whether the configuration file was correctly imported.

Format	Field	Description
%c%c	Cmd	Xx
%1x	Success/ Failure	0 – The file could not be successfully imported 1 – The file was successfully imported

For the configuration settings to take effect, once the configuration file has been successfully loaded (as indicated by the response) the software needs to be restarted.

Audio Commands

These automation commands have been developed to support the Easysound audio model. This enables users to control audio with greater flexibility than with earlier systems (as described on page 75). Care should be taken to ensure that the two control methods are not used together.

All commands which address this system are preceded by a lower case 'j'. Further parameters will allow the user to control:

- Audio inputs
- Audio outputs
- Voice-overs
- Transitions
- Audio preview

A full set of enquiry commands are also included to enable feedback of the audio parameters.

At various places in the Easysound audio model commands, gain values in decibels (dB) are sent to and from the Imagestore. These are represented as a number from 0 to 255, sent as 2 hex digits.

A floating point dB value is converted to an integer in the following way:

```
int dBVal(float dB) // dB is in range -99.5 to 28
{
    int val = (dB * 2) + 199;

    if (val < 0)
    {
        val = 0;
    }
    else if (val > 255)
    {
        val = 255;
    }

    // printf("val of dB %f is: %d", dB, val);
    return val;
}
```

Inputs **j0**

Input parameters can be changed remotely to suit applications. Channels can be adjusted, phase inverted, shuffled and selected via the command strings listed below.

Control Input 'j0'	Input Number	Function	Function Parameters
--------------------	--------------	----------	---------------------

The command *j0* instructs the Oxtel unit that a parameter of an audio input is to be changed. The first parameter to be sent is the audio input number. Each input consists of 4 channels (equivalent to one group or two AES pairs).

The input number 1, 2 or 3 relates to the physical inputs of the Easysound such that:

Input	A/B Mix Mode	Two Voice-over mode
1	A Input	Background Input
2	VO Input	Voice-over 1
3	B Input	Voice-over 2

The second parameter is the function that has to be performed.

Function	Function CMD
Shuffle	0
Mute	1
Trim	2
Phase	3
Level	4
Source	5

Several functions have additional parameters *Channel/Pair/Group*, *Channel*. These allow the user to assign values to several channels of audio at once.

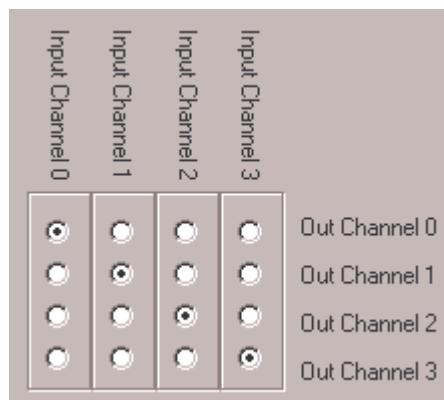
If the '*Channel/Pair/Group*' parameter is given a value of 1, only one channel (indicated by 'Channel Number') will be set.

If the *Channel/Pair/Group* parameter is given the value 2, a stereo pair will be affected. In this case the *Channel Number* dictates which pair is set, Channel Number = 0 for the first pair and Channel Number = 2 for the second pair.

If *Channel/Pair/Group* is given a value of 4, then the whole group (4 channels) is affected and the *Channel Number* parameter is ignored.

Shuffle **j0 0**

The shuffle values allow mono input channels to be moved around within a group. In this way pair swaps can be achieved. Each output channel can take its input from any one of the input channels. Each input can be copied to multiple output channels.



CMD	Input	Function	Param_1	Param_2
j0	%01x Input	%01x Shuffle	%01x Channel In	%01x Channel Out
j0	1,2 or 3	0	0,1,2 or 3	0,1,2 or 3

```
void SetInputShuffle(int Input, int ChannelIn,
                     int ChannelOut);
{
    remote_send("j0%01x0%01x%01x", Input,
                ChannelIn, ChannelOut);
}
```

Mute j0 1

Channels, pairs or groups can be muted with one command.

CMD	Input	Function	Param_1	Param_2	Param_3
j0	%01x Input	%01x Mute	%01x Channel/ Pair/Grou p	%01x Channel	%01x Mute On/Off
j0	1, 2 or 3	1	1, 2 or 4	0, 1, 2 or 3	0 =Mute On 1= Mute Off

```
void SetInputMute(int Input, int ChannelPairGroup,
                  int Channel, int MuteOnOff);
{
    remote_send("j0%01x1%01x%01x%01x", Input,
                ChannelPairGroup, Channel,
                MuteOnOff);
}
```

Trim j0 2

Channels, pairs or groups can be trimmed with one command. The trim allows individual channels to be boosted or attenuated by a range of -100 dB to $+28$ dB. This is helpful for quiet audio sources. The adjustment of trim occurs after the ‘input level’.

To enable $\frac{1}{2}$ dB resolution of the level value, a range of 0 to 255 (0x00 to 0xFF) is used such that 0 gives a -100.0 db attenuation, 255 gives a $+28$ dB gain, it follows therefore that a value of 199 (0xC7) is equivalent to 0dB.

Whilst the large range for trim adjustment exists, it is recommended that only small values (-6 dB to $+6$ dB) are used.

CMD	Input	Function	Param_1	Param_2	Param_3
j0	%01x Input	%01x Trim	%01x Channel/Pair/Group	%01x Channel	%02x Trim
j0	1, 2 or 3	2	1, 2 or 4	0, 1, 2 or 3	dBVal

```

void SetInputTrim(int Input, int ChannelPairGroup,
                  int Channel, float Trim);
{
    remote_send("j0%01x2%01x%01x%02x ", Input,
                ChannelPairGroup, Channel,
                dBVal(Trim));
}

```

Phase **j0 3**

Channels, pairs or groups can be phase inverted with one command.

CMD	Input	Function	Param_1	Param_2	Param_3
j0	%01x Input	%01x Phase Invert	%01x Channel/Pair/Grou p	%01x Channel	%01x Phase Invert
j0	1, 2 or 3	3	1, 2 or 4	0, 1, 2 or 3	0 / 1

```

void SetInputPhase(int Input, int ChannelPairGroup,
                   int Channel, int PhaseInvert);
{
    remote_send("j0%01x3%01x%01x%01x ", Input,
                ChannelPairGroup, Channel,
                PhaseInvert);
}

```

A phase invert value of zero enables the audio to pass unaffected whilst a phase invert value of 1 inverts the wave form.

Level **j0 4**

Input levels for channels, pairs or groups can be adjusted with one command. The level range is -99dB to +28dB, with a special case of minus infinity dB. To enable $\frac{1}{2}$ dB resolution of the level value, a range of 0 to 255 (0x00 to 0xFF) is used such that 0 gives a -infinity dB attenuation. 1 gives -99dB.attenuation, 255 gives a +28dB gain, it follows therefore that a value of 199 (0xC7) is equivalent to 0dB.

CMD	Input	Function	Param_1	Param_2	Param_3
j0	%01x Input	%01x Level	%01x Channel/Pair/Group	%01x Channel	%02x Level
j0	1, 2 or 3	4	1, 2 or 4	0, 1, 2 or 3	dbVal

```

void SetInputLevel(int Input,
                   int ChannelPairGroup,
                   int Channel, float Level);
{
    remote_send("j0%01x4%01x%01x%02x", Input,
                ChannelPairGroup, Channel,
                dbVal(Level));
}

```

Input Source**j0 5**

The *A/BGND* input and the *B/Voice-over 2* input have the ability to pick audio from either the incoming AES streams or the audio embedded with the video inputs.

Parameter 2, the source, is represented in this table:

Physical Audio Source	Source Number
Group 1	0
Group 2	2
Group 3	4
Group 4	6
AES	8

CMD	Input	Function	Param_1	Param_2
j0	%01x Input	%01x Source	%01x Pair	%01x Source
	1, 2 or 3	5	1 or 2	0, 2, 4, 6 or 8

```

void SetInputTrim(int Input, int Pair,
                  int Source);
{
    remote_send("j0%01x%01x%01x", Input, Pair,
                Source);
}

```

Outputs j1

There are two outputs from the Easysound audio mixer; Program and Preview. The preview output allows the operator to monitor various audio mixes and levels before taking the desired effect ‘to air’. The preview output can also be used for effects send and return applications.

The program output is the main audio output. As with the audio inputs, levels can be adjusted. It is also possible to embed the audio to one of the four groups of the outgoing video.

All output command strings start with the string ‘j1’.

Output Level j1 0

Output levels for channels, pairs or groups can be adjusted with one command. The level range is -99dB to +28d, with a special case of minus infinity dB. To enable ½ dB resolution of the level value, a range of 0 to 255 (0x00 to 0xFF) is used such that 0 gives a -infinity dB attenuation. 1 gives -99.0db attenuation, 255 gives a +28dB gain, it follows therefore that a value of 199 (0xC7) is equivalent to 0dB.

CMD	Output	Function	Param_1	Param_2	Param_3
j1	%01x 0	%01x Level	%01x Channel / Pair / Group	%01x Channel	%02x Level
	0 = PGM	0	1, 2 or 4	0, 1, 2 or 3	dbVal

```

void SetOutputLevel(int ChannelPairGroup,
                     int Channel, float Level);
{
    remote_send("j10%01x%01x%02x",
                ChannelPairGroup, Channel,
                dbVal(Level));
}

```

Output Level via Gain Type j1 3

In Imagestore 750 and version 1.13.5 of Imagestore 300[+] and Imagestore HD-TV, extended output gain control is introduced for all audio channels (embedded and AES) – not just channels within the first group.

The mixfile command g (see page 79) is used to associate a set of audio channels with an output gain type (for example, PROGRAM_OUTPUT). The output gain level for this set of channels may then be controlled via the j103 automation command.

CMD	Output	Function	Param_1	Param_2
j1	%01x 0	%01x via Gain Type	%02x Output Gain Type	%02x Level
	Unused	3	0 = PROGRAM_GAIN	dbVal

```
void SetOutputLevelGainType( int GainType,
                           float Level );
{
    remote_send( "j103%02x%02x", GainType,
                 dbVal(Level));
}
```

Example automation commands for PROGRAM_GAIN:

j1030000: -infinity dB
j1030001: -99.0 dB
j10300C7: 0.0 dB
j10300FF: +28.0 dB

Output Mute j1 1

Channels, pairs or groups can be muted with one command.

CMD	Output	Func	Param_1	Param_2	Param_3
j1	%01x 0	%01x Mute	%01x Channel / Pair / Group	%01x Channel	%01x Mute On/Off
	0 = PROGRAM	1	1, 2 or 4	0, 1, 2 or 3	0 = Mute On 1 = Mute Off

```

void SetInputMute(int ChannelPairGroup,
                  int Channel, int Mute);
{
    remote_send( "j101%01x%01x%01x" ,
                  ChannelPairGroup, Channel, Mute);
}

```

Output Embed **j1 2**

The outgoing audio can be embedded to one of the four audio groups in the output video. The group number can be selected with the following string.

CMD	Output	Function	Param_1
j1	%01x 0	%01x Embed	%01x Group
j1	0 = PROGRAM	2	0, 2, 4 or 6

The group in which the audio is embedded is reflected by the value of Param_2.

Physical Audio Source	Param_1 Value
Group 1	0
Group 2	2
Group 3	4
Group 4	6

```

void SetOutputEmbed(int Group);
{
    remote_send( "j102%01x" , (Group -1) * 2 );
}

```

Voice-Overs **j2**

The Easysound enables the user to insert voice-overs into their normal audio program, a variety of configurations are available.

Imagestore 2/3

A/B mix mode provides two stereo pair voice-overs that can be inserted into the result of the mix. The voice-over in the A/B mix mode is named Voice-Over 1.

Two voice-over mode provides four stereo pair voice-overs that can be inserted into the background or program audio. The Voice-overs in Two voice-over mode are named Voice-Over 1 and Voice-Over 2. This mode is rarely used since one SDI input is lost to provide inputs for the second voice-over.

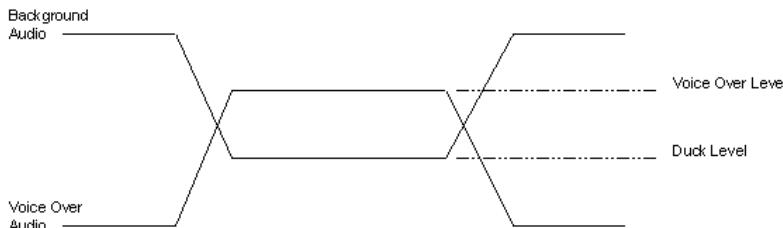
Imagestore 300[+] and Imagestore HD-TV

There are two voice-overs (one of which may be an Easyplay voice-over). Each voice-over which may contain multiple channels. There is therefore no concept of being able to control separate stereo pairs within a single voice-over like with Imagestore 2/3.

Each voice-over has two parameters that need to be considered. Firstly the 'Duck Level'. Duck level is the amount by which the background audio is attenuated by when a voice-over becomes active. This level is measured in dB. For example, when the voice-over is on, the background 'ducks' by –12dB.

The second parameter to consider is the voice-over preset. This represents the level of the voice-over audio when the voice-over becomes active. For example, when the voice-over is on, the voice-over audio comes up to 0dB.

These levels are represented by the following diagram:



Duck Preset j2 0

This sets the attenuation of the background audio when the voice-over is on.

CMD	Voice-over	Function	Param_1	Param_2	Param_3
j2	%01x Voice-over	%01x Duck Preset	%01x Channel/Pair /Group	%01x Channel	%02x Duck Level
j2	1 or 2	0	1, 2 or 4	0,1,2 or 3	dBVal

```
void SetDuck(int VO, int ChannelPairGroup,
             int Channel, float Duck);
{
    remote_send("j2%01x0%01x%01x%02x ", VO,
                ChannelPairGroup, Channel,
                dBVal(Duck));
}
```

Voice-over Preset j2 1

This sets the level of the voice-over audio when the voice-over is on.

CMD	Voice-over	Function	Param_1	Param_2	Param_3
j2	%01x Voice-over	%01x Voice Preset	%01x Channel/Pair/Group	%01x Channel	%02x Voice Level
j2	1 or 2	1	1, 2 or 4	0,1,2 or 3	dBVal

```
void SetDuck(int VO, int ChannelPairGroup,
             int Channel, int Float);
{
    remote_send("j2%01x1%01x%01x%02x ", VO,
                ChannelPairGroup, Channel,
                dBVal(Voice));
}
```

Voice-over Level j2 2

This command allows the voice-over to be directly turned on, off or set to an intermediate position.

Normally the voice-over would be controlled by the fade/cut voice-over transition commands, but this command allows voiceovers to be controlled via a T-bar or external fader. With Level = 0 the voiceover is off, and at 255 (0xFF) the voiceover is fully on – (the VO input will have reached its preset level and the background will be ducked to the duck level).

CMD	Voice-over	Function	Param_1	Param_2	Param_3
j2	%01x Voice-over	%01x Voice-over	%01x Channel/Pair/Group	%01x Channel	%02x Voice-over Level
j2	1 or 2	2	1, 2 or 4	0,1,2 or 3	0 to 255 (0x00-0xFF)

```
void SetDuck(int VoiceOver, int ChannelPairGroup,
             int Channel, int VoiceOverLevel);
{
    remote_send("j2%01x2%01x%01x%02x ", VoiceOver,
                ChannelPairGroup, Channel,
                VoiceOverLevel);
}
```

Rates j3

The Easysound can perform any automatic transitions. Cutting and fading mixes, voice-overs and output levels. The transition rate (or duration) can be set remotely. Transition times are in fields (not frames). A maximum transition rate of 600 fields gives a duration of 10 seconds for a 30 frames per second system and a duration of 12 seconds for a 25 frames per second system.

Fade to/from Silence **j30**

CMD	Function	Param_1
j3	%01x Transition	%03x Duration in fields
j3	0	0 to 600 (0x000 to 0x258)

```
void SetSilenceRate(int fields);
{
    remote_send("j30%03x", fields);
}
```

Fade A/B **j31**

CMD	Function	Param_1
j3	%01x Transition	%03x Duration in fields
j3	1	0 to 600 (0x000 to 0x258)

```
void SetABRate(int fields);
{
    remote_send("j31%03x", fields);
}
```

Fade Voice-over 1 Pair 1 **j32**

CMD	Function	Param_1
j3	%01x Transition	%03x Duration in fields
j3	2	0 to 600 (0x000 to 0x258)

```
void SetV01Rate(int fields);
{
    remote_send("j32%03x", fields);
}
```

Note: For Imagestore 300[+] and Imagestore HD-TV this command has different usage since it is used to fade all channels in voice-over 1. It should therefore be named 'Fade Voice-over 1'.

Fade Voice-over 1 Pair 2 j33

CMD	Function	Param_1
j3	%01x Transition	%03x Duration in fields
j3	3	0 to 600 (0x000 to 0x258)

```
void SetVO2Rate(int fields);
{
    remote_send("j33%03x", fields);
}
```

Note: For Imagestore 300[+], Imagestore HD-TV and Imagestore 750 this command has different usage since it is used to fade all channels in voice-over 2. It should therefore be named 'Fade Voice-over 2'.

Fade Voice-over 2 Pair 1 j34

CMD	Function	Param_1
j3	%01x Transition	%03x Duration in fields
j3	4	0 to 600 (0x000 to 0x258)

```
void SetVO1Rate(int fields);
{
    remote_send("j34%03x", fields);
}
```

Note: For Imagestore 300[+], Imagestore HD-TV and Imagestore 750 this command is unsupported.

Fade Voice-over 2 Pair 2 j35

CMD	Function	Param_1
j3	%01x Transition	%03x Duration in fields
j3	5	0 to 600 (0x000 to 0x258)

```
void SetVO2Rate(int fields);
{
    remote_send("j35%03x", fields);
}
```

Note: For Imagestore 300[+], Imagestore HD-TV and Imagestore 750 this command is unsupported.

Transitions j4

The Easysound can perform many automatic transitions. Cutting and fading mixes, voice-overs and output levels.

Cut A/B j40

When in A/B mix mode, cuts between the A and B audio. If the destination is A then Param_1 takes the value of 0. If the destination is B then Param_1 should be 1.

CMD	Function	Param_1
j4	%01x Transition	%01x Destination A/B
j4	0	0 or 1

```
void ABCut(int Destination);
{
    remote_send("j40%01x", Destination);
}
```

Fade A/B **j41**

When in A/B mix mode, fades between the A and B audio. If the destination is A then Param_1 takes the value of 0. If the destination is B then Param_1 should be 1.

CMD	Function	Param_1
j4	%01x Transition	%01x Destination A/B
j4	1	0 or 1

```
void ABFade(int Destination);
{
    remote_send("j41%01x", Destination);
}
```

Cut to/from Silence **j42**

Cuts the audio output to and from silence. If cutting to silence then Param_1 takes the value of 0. If cutting from silence then Param_1 should be 1.

CMD	Function	Param_1
j4	%01x Transition	%01x To/From Silence
j4	2	0 or 1

```
void SilenceCut(int Silence);
{
    remote_send("j42%01x", Silence);
}
```

Fade to/from Silence **j43**

Fades the audio output to and from silence. If fading to silence then Param_1 takes the value of 0. If fading from silence then Param_1 should be 1.

CMD	Function	Param_1
j4	%01x Transition	%01x To/From Silence
j4	3	0 or 1

```
void SilenceFade(int Silence);
{
    remote_send("j43%01x", Silence);
}
```

Fade Voice-over 1 From/To Silence j44

Fades voice-over one up and down. Voice-over pairs can be controlled independently so a ‘pairs’ parameter is provided. Parameter 2 dictates the destination of the transition, 0 to fade the voice-over down and 1 to fade the voice-over up.

CMD	Function	Param_1	Param_2
j4	%01x Transition	%01x Pair	%01x To/From Silence
j4	4	1 or 2	0 or 1

```
void VO1Fade(int Pair, int Destination);
{
    remote_send("j44%01x%01x", Pair, Destination);
}
```

Note: For Imagestore 300[+], Imagestore HD-TV and Imagestore 750 this command has different usage.

For Imagestore 300[+], Imagestore HD-TV and Imagestore 750, this command is used to fade both voice-over 1 and voice-over 2 to or from silence, with parameter 1 determining the number of the voice-over to be faded, as follows:

CMD	Function	Param_1	Param_2
j4	Transition	Voice-over	To/From Silence
j4	4	1 or 2	0 or 1

```
void VO1Fade(int Destination);
{
    remote_send("j441%01x", Destination);
}
```

```
void VO2Fade(int Destination);
{
    remote_send("j442%01x", Destination);
}
```

Fade Voice-over 2 From/To Silence **j45**

As for voice-over 1.

CMD	Function	Param_1	Param_2
j4	%01x Transition	%01x Pair	%01x To/From Silence
j4	5	1 or 2	0 or 1

```
void VO2Fade(int Pair, int Destination);
{
    remote_send("j45%01x%01x", Pair, Destination);
}
```

Note: For Imagestore 300[+], Imagestore HD-TV and Imagestore 750 this command is unsupported. Both VO1 and VO2 are faded using the “j4 4” command.

Cut Voice-over 1 **j46**

Cuts voice-over one up and down. Voice-over pairs can be controlled independently so a ‘pairs’ parameter is provided. Parameter 2 dictates the destination of the transition, 0 to cut the voice-over down and 1 to cut the voice-over up.

CMD	Function	Param_1	Param_2
j4	%01x Transition	%01x Pair	%01x To/From Silence
j4	6	1 or 2	0 or 1

```
void VO1Cut(int Pair, int Destination);
{
    remote_send("j46%01x%01x", Pair, Destination);
}
```

Note: For Imagestore 300[+], Imagestore HD-TV and Imagestore 750 this command has different usage.

For Imagestore 300[+], Imagestore HD-TV and Imagestore 750, this command is used to cut both voice-over 1 and voice-over 2 to or from silence, with parameter 1 determining the number of the voice-over to be faded, as follows:

CMD	Function	Param_1	Param_2
j4	%01x Transition	%01x Voice-over	%01x To/From Silence
j4	6	1 or 2	0 or 1

```
void VO1Cut(int Destination);
{
    remote_send("j461%01x", Destination);
}
```

```
void VO2Cut(int Destination);
{
    remote_send("j462%01x", Destination);
}
```

Cut Voice-over 2 j47

As for voice-over 1.

CMD	Function	Param_1	Param_2
j4	%01x Transition	%01x Pair	%01x To/From Silence
j4	7	1 or 2	0 or 1

```
void VO2Cut(int Pair, int Destination);
{
    remote_send("j47%01x%01x", Pair, Destination);
}
```

Note: For Imagestore 300[+], Imagestore HD-TV and Imagestore 750 this command is unsupported. Both VO1 and VO2 are cut using the “j4 6” command.

Audio Follow Video **j5**

Commands can be sent to the Easysound such that audio levels follow video transitions.

Audio Follow Fade to Black **j50**

The main audio output level follows the output video ‘fade to black’ state. If the program video is faded to black, the audio will fade to silence.

CMD	Function	Param_1
j5	%01x Follow	%01x On/Off
j5	0	1 or 0

```
void FollowFTB(int OnOff);
{
    remote_send( "j50%01x", OnOff );
}
```

Audio Follow A/B **j51**

The audio from the A and B inputs follows the video from the A and B inputs. If the video does a cross fade, V fade, cut etc. the audio follows.

CMD	Function	Param_1
j5	%01x Follow	%01x On/Off
j5	1	1 or 0

```
void FollowAB(int OnOff);
{
    remote_send( "j51%01x, OnOff );
}
```

Audio Follow Keyers j52

The audio output level follows the output video Keyer states. If a keyer does a cross fade, V fade, cut, etc., the audio in that Keyer follows. While disabled, the audio will operate independently of the Keyer.

CMD	Function	Param_1
j5	%01x Follow	%01x On/Off
j5	2	1 or 0

```
void FollowDSK(int OnOff);
{
    remote_send( "j52%01x, OnOff");
}
```

Voice-over Follow Easyplay j53

Activates voice-over 1 whenever an Easyplay file is playing out.

CMD	Function	Param_1
j5	%01x Follow	%01x On/Off
j5	3	1 or 0

```
void FollowDSK(int OnOff);
{
    remote_send( "j52%01x, OnOff");
}
```

Audio Preview j6

There are 2 outputs from the Easysound audio mixer, Program and Preview. The preview output allows the operator to monitor various audio mixes and levels before taking the desired effect 'to air'. The preview output can also be used for effects send and return applications.

Preview selections are:

Preview	Preview Value
A Input	0x00
B Input	0x01
A/B Mix	0x02
Voice Over	0x03
Voice Over Enabled	0x04
Program	0x05
Silence	0x06
Test Tone	0x07
2 Voice-over	0x08
Full A/B Mix	0x09

Preview selections for Imagestore 750 are:

Preview	Preview Value
Embedded Input	0xFF
A Input	0x00
B Input	0x01
A/B Mix	0x02
Voice Over Input	0x03
Voice Over Mix	0x04
Program	0x05
Silence	0x06
Test Tone	0x07
Passthrough	0x08

The command string is structured:

CMD	Param_1
j6	%02x Preview

```
void SelectPreview(int Preview);
{
    remote_send( "j6%02x, Preview");
}
```

The preview output is AES only and is not embedded into the output video stream.

Audio Enquire j7

The internal parameters and settings of Easysound can be interrogated for feedback and tracking purposes.

All parameters are returned in a string containing a number of fixed-width parameters, without commas or other separating characters. The width of each parameter field is listed in the tables below.

Enquire Inputs j70

Input parameters can be requested using:

CMD	Function	Param_1
j7	0	%01x Input

```
void EnquireInputParameters(int Input)
{
    remote_send( "j70%01x", Input );
}
```

The information returned is:

Format	Field	Description
%c	j	Audio
%c	7	Enquire
%c	0	Input
%c	Input Number	1, 2 or 3
%02x	Shuffle 0	0, 1, 2 or 3
%02x	Shuffle 1	0, 1, 2 or 3
%02x	Shuffle 2	0, 1, 2 or 3
%02x	Shuffle 3	0, 1, 2 or 3
%02x	Input Level Pair 1	dBval

Format	Field	Description
%02x	Input Level Pair 2	dBval
%02x	Input Trim Pair 1	dBval
%02x	Input Trim Pair 2	dBval
%02x	Input Source Pair 1	0, 2, 4, 6 or 8 (Only returned for inputs 1 and 3)
%02x	Input Source Pair 2	0, 2, 4, 6 or 8 (Only returned for inputs 1 and 3)
%01x	Phase Invert Pair 1	0 or 1
%01x	Phase Invert Pair2	0 or 1
%01x	Input Mute Pair 1	0 or 1
%01x	Input Mute Pair 2	0 or 1

Enquiring Inputs (j70) now returns the 4-bit shuffle mask values for each input channel as set by Set Audio Input Shuffles (jg) see page 191.

Enquire Voice-over j71

Output parameters can be requested with the following example.

CMD	Function	Param_1
j7	%01x	%01x
	1	Voice over

```
void EnquireVoiceOverParameters(int VoiceOver)
{
    remote_send("j71%01x", VoiceOver );
}
```

The information returned is:

Format	Field	Description
%c	j	Audio
%c	7	Enquire
%c	1	Voice-over
%c	Voice-over Number	1, 2, 3 or 4
%02x	Voice-over Preset	dBval
%02x	Duck Preset	dBval
%02x	Voice-over Level	0 to 255 (0x00 to 0xFF)

Where voice-over number is:

Voice-over	Voice-over Number
Voice-over 1 Pair 1	1
Voice-over 1 Pair 2	2
Voice-over 2 Pair 1	3
Voice-over 2 Pair 2	4

Enquire Outputs j72

Output parameters can be requested with the following example.

CMD	Function	Param_1
j7	%01x	%01x
	2	Output

```
void EnquireOutputParameters(int Output)
{
    remote_send("j72%01x", Output );
}
```

The information returned is:

Format	Field	Description
%c	j	Audio
%c	7	Enquire
%c	2	Output
%c	Output Number	1 (Program Output)
%02x	Output Level Pair 1	dBval
%02x	Output Level Pair 2	dBval
%01x	Output Mute Pair 1	0 or 1
%01x	Output Mute Pair 2	0 or 1
%01x	Output Embed Pair 1	0, 2, 4 or 6 – for output 1 only.
%01x	Output Embed Pair 2	0, 2, 4 or 6 - for output 1 only.

Enquire Rates **j73**

Transition durations can be requested with the following example.

CMD	Function
j7	%01x 3

```
void EnquireRates()
{
    remote_send( "j73" );
}
```

The information returned is:

Format	Field	Description
%c	j	Audio
%c	7	Enquire
%c	3	Rates
%03x	Fade to silence duration	0 to 600
%03x	A/B Mix duration	0 to 600
%03x	Voice-over 1 Pair 1 fade duration ⁶	0 to 600
%03x	Voice-over 1 Pair 2 fade duration ⁷	0 to 600
%03x	Voice-over 2 Pair 1 fade duration ⁸	0 to 600
%03x	Voice-over 2 Pair 2 fade duration ⁹	0 to 600

Transition durations are returned in fields.

Enquire Audio Follow Video **j74**

The audio follow video settings can be requested from the Oxtel unit.

CMD	Function
j7	%01x 4

```
void EnquireAudioFollow()
```

⁶ Voice-over 1 for Imagestore 300, Imagestore HD-TV and Imagestore 750.

⁷ Voice-over 2 for Imagestore 300, Imagestore HD-TV and Imagestore 750.

⁸ Redundant for Imagestore 300, Imagestore HD-TV and Imagestore 750.

⁹ Redundant for Imagestore 300, Imagestore HD-TV and Imagestore 750.

```
{
    remote_send( "j74" );
}
```

The information returned is:

Format	Field	Description
%c	j	Audio
%c	7	Enquire
%c	4	Audio Follow
%01x	Audio follow fade to black	0 or 1
%01x	Audio follow A/B mix	0 or 1

Where 0 indicates that audio follow is inactive and 1 indicates that audio follow is active.

Enquire Preview j75

Returns the current preview mode setting.

CMD	Function
j7	%01x 5

```
void EnquirePreview()
{
    remote_send( "j75" );
}
```

The information returned is:

Format	Field	Description
%c	j	Audio
%c	7	Enquire
%c	5	Preview
%02x	Preview Mode	See below

Preview Mode	Return Value
A Input	0
B Input	1

Preview Mode	Return Value
Mix	2
Voice-over 1 Pair 1	3
Voice-over 1 Pair 2	4
Voice-over 2 Pair 1	5
Voice-over 2 Pair 2	6
Voice-over on	7
Program Copy	8
Silence	9

For Imagestore 750:

Preview Mode	Return Value
A Input	0
B Input	1
A/B Mix	2
Voice-over Input	3
Voice-over Mix	4
Program	5
Silence	6
Test Tome	7
Passthrough	8

Enquire Clean Feed **j76**

Returns the current clean feed setting in the same form as the j75 command.

CMD	Function
j7	%01x 6

```
void EnquireCleanFeed( )
{
    remote_send( "j76" );
}
```

Enquire Monitor j77

Returns the current monitor setting in the same form as the j75 command.

CMD	Function
j7	%01x 7

```
void EnquireMonitor()
{
    remote_send( "j77" );
}
```

Enquire Output Gain j79

Returns the current output mode gain setting.

CMD	Function
j7	%01x 9

```
void EnquireOutputGain()
{
    remote_send( "j79" );
}
```

The information returned is:

Format	Field	Description
%c	J	Audio
%c	7	Enquire
%c	9	Output Gain
%01d	Output Gain	0 = PROGRAM_GAIN
%02x	Level	Gain level (dbVal)

Audio FX Send/Return and Data Paths

FX Send Setup j8

Sets up the audio FX send and return paths, and any gain applied to the FX send path.

CMD	Param_1	Param_2	Param_3	Param_4
j8	%1x Enable	%2x: FX Send mode.	%2x: FX Return pair	%2x: FX Send gain.
j8	0 = Disabled, 1 = Enabled	See the tables below	4 = Pair1, 6 = Pair2	dBval

For AB Mix mode, values for send mode are as follows:

FX Send mode	Value
A Input Pair 1	1
A Input Pair 2	2
Voice-over 1 Pair 1	4
B Input Pair 1	8
B Input Pair 2	16
AB Mix Pair 1	9
AB Mix Pair 2	18
AB Mix Pair 1 + VO1	13
AB Mix Pair 2 + VO1	22

And for dual voice-over mode, the following values are allowed:

FX Send mode	Value
BGND Pair 1	1
BGND Pair 2	2
Voice-over 1 Pair 1	4
Voice-over 2 Pair 1	8
Voice-over 2 Pair 2	16
BGND + VO1 Pair 1	5
BGND + VO2 Pair 1	13
BGND + VO2 Pair 2	29

Example

```
void Set_FX_Send(bool Enable, int SendMode,
                  int ReturnPair, float SendGain)
{
    remote_send( "j8%01x%02x%02x%02x" , Enable,
                  SendMode, ReturnPair, SendGain);
}
```

Audio Data Path Setup j9

Specifies that a particular audio pair contains a data-stream rather than simple PCM audio data. This is used to enable a data ‘pass-through’ for systems such as Dolby E and Dolby Digital. When using this, the audio data is guaranteed to be passed unmodified, and cuts rather than fades are applied to the data values.

CMD	Param_1	Param_2	Param_3
j9	%1x Enable: 0 = Data path disable 1 = Data path enabled	%02x Input data Pair	%02x Output data Pair

Enhanced AB Mixer Controls

Various additional commands for controlling the Easysound AB audio mixer have been added to support Presmaster functions. These are documented here.

Audio AB Position **ja**

Directly sets up the ‘position’ of the AB mix fader. Used for T-bar control of AB mixer.

CMD	Param_1
ja	%03x Mix position. 0 = Fader at B, 512 (or 0x200) = Fader at A

Audio AB Mix Mode **jb**

Switches between audio cross-fades and audio v-fades.

CMD	Param_1
jb	%01x Fade mode 0 = X-fades, 1 = V-fades (through silence)

Audio AB Asymmetric Transition **jc**

Use this to perform an asymmetric audio transition (such as a V-fade) with a single call.

Rate 1 defines the duration of the first half of the fade (time take to reach the midpoint), while Rate 2 is the time from the midpoint to completion. Both rates are measured in fields. Note, only Rate 1 is used if **not** doing a V-fade.

CMD	Param_1	Param_2	Param_3
jc	%01x Direction. 0 = To A, 1 = To B	%03x: Rate 1 in fields	%03x: Rate 2 in fields

Audio AB Fade to Position jd

Like the audio AB position command ('ja') but allows a fade the specified position instead of just a cut.

CMD	Param_1	Param_2
jd	%03x Mix position. 0 = Fader at B, 512 (or 0x200) = Fader at A	%03x: Rate in fields.

Audio A and B Fade Position je & jf

These commands allow the two halves of the audio AB mixer to be controlled independently.

Instead of treating the A fader and B fader as two linked faders, where one is the inverse of the other, this command allows the A fader to be manipulated independently. This can be used to perform effects such as voiceovers in the AB mixer. You might have the voiceover source presented on the B input, and the program audio coming from A. In this case, assuming the AB mixer is cut to A, you would simply fade the B up to the required level and duck the A down.

CMD	Param_1	Param_2
je - A position jf - B position	%03x Mix position. 0 = Fully off, 512 (or 0x200) = Fully on	%03x: Rate in fields.

Note that the fader 'positions' are in linear gain values, rather than expressed in dB.

Set Audio Input Shuffles jg

CMD	Param_1	Param_2	Param_3
jg	%1x Input 1, 2 or 3	%1x Channel In 0,1,2 or 3	%1x Mask See below

The Mask is a 4-bit value. Each set bit will connect the respective output channel to the specified input channel.

- Bit 0 - Channel Out 0
- Bit 1 - Channel Out 1
- Bit 2 - Channel Out 2
- Bit 3 - Channel Out 3

For example a mask value of 10, 0xa or 1010b would connect output channels

1 and 3 to the specified input channel.

This command should be used in preference to the j0 0 command which only permits one output channel to be connected to each input channel.

Audio Clean Feed Select jh

This command sets the clean feed audio monitoring point for Imagestore 750. It behaves in the same way as j6, except that it operates on the clean feed.

CMD	Param_1
jh	%02x: Preview (as defined for Imagestore 750 in command j6)

Audio Monitor Feed Select ji

This command sets the monitor feed audio monitoring point for Imagestore 750. It behaves in the same way as j6, except that it operates on the monitor feed.

CMD	Param_1
jh	%02x: Preview (as defined for Imagestore 750 in command j6)

Setup Audio Delay Bank jl

This command sets the DVE compensation enable and delay for a delay bank.

CMD	Param_1	Param_2	Param_3
jl	%1x Delay bank No.	%1x DVE Compensation Enabled 0 – Disabled, 1 – Enabled	%06x Delay (in samples)

Audio Metering Commands

Set Audio Metering Mode Y0

Specify audio metering mode, and specifies which mixes are to be metered.

On start-up, metering defaults to off. Once metering is enabled, the audio Metering packets described below are sent on a regular basis.

Metering of up to 4 pairs is supported, with each pair being selectable individually.

The ‘pairs to meter’ parameter is currently ignored, and Output pairs 1 to 4 are always transmitted –assuming the metering mode is not **off**.

CMD	Param_1	Param_2
Y0	%1x: Metering Mode	%04x: Pairs to meter – currently ignored.

Example:

```
void MeterInputs()
{
    // Enable metering on all 4 pairs at
    // half frame rate.
    remote_send("Y011234");
}
```

The metering mode specifies the rate at which metering packets will be sent

Metering Modes:

Off	0
Meter at half frame rate	1
Meter at frame rate	2

Audio Metering Packet Y0

An unsolicited packet, sent every frame or field when audio metering is enabled (see above)

CMD	Param_1
Y0	<p>%8c: Metering values. These are all expressed in dB. Each byte represents a meter value from a single mono input.</p> <p>Param_1 is a string of 8 bytes, one for each output being metered</p>

All values all have the top bit set. To convert a value to dB, use the following code:

```
unsigned char inputchar;
int dB;
dB = inputchar - (255);
```

The values sent are the dB value added to 255. This allows a metering range of -254dB (1) to 0dB (255) in 1 dB steps.

When a channel has metering turned OFF no data bytes are sent for that channel sent, reducing the packet length. If all metering is turned off then no metering packets are sent.

Dolby and Up-mix Commands

Note: The Dolby and up-mix commands are applicable to Imagestore 750 v2.0 or above with the relevant Dolby or up-mix module fitted. See the Imagestore 750 user manual for more information.

Set Up-Mix Mode q00

This command sets the operating mode of an up-mix module.

CMD	Param_1	Param_2	Param_3
q0	%1x: 0	%1x: Up-Mix Module No.	%1x: Mode 0 – Up-mix pass-through 1 – Up-mix on 2 – Up-mix controlled by metadata

Set Dolby E Encoder Mode q10

This command sets the operating mode of a Dolby E encoder module.

CMD	Param_1	Param_2	Param_3
q1	%1x: 0	%1x: Dolby E Encoder Module No.	%1x: Mode 0 – Encode 1 – Pass-through

Set Dolby Digital Encoder Mode q20

This command sets the operating mode of a Dolby E encoder module.

CMD	Param_1	Param_2	Param_3
q2	%1x: 0	%1x: Dolby Digital Encoder Module No.	%1x: Mode 0 – Encode 1 – Pass-through

Set Dolby Decoder Mode q30

This command sets the operating mode of a Dolby E encoder module.

CMD	Param_1	Param_2	Param_3
q3	%1x: 0	%1x: Dolby Decoder Module No.	%1x: Mode 0 – Automatic 1 – Dolby E 2 – Dolby D 3 – Mute

Metadata Commands

Note: The metadata commands are applicable to Imagestore 750 v2.0 or above. See the Imagestore 750 user manual for more information.

Note that these commands have a “Flags” parameter that are not yet used and are reserved for future use. The value “0” is currently to be used for this parameter for all commands.

Set Pitch Shift Code o0

This command sets the pitch shift code, to indicate the amount of pitch shift between the original and current playback speeds of the audio frame. See the Dolby documentation for how to calculate the pitch shift code.

CMD	Param_1	Param_2	Param_3
o0	%1x: Bank No.	%1=02x: Flags 00	%08x: Pitch shift code 0 – No pitch shift) (0x-800 to 0x07FF)

The current up-mix mode can be enquired using:

CMD	Param_1
q0	%1x: Bank No.

The response is in the form of the set command.

Set Program Description o1

This command sets the description of a program using any ASCII characters between 0x20 and 0x7E. This command is not applicable to user metadata.

CMD	Param_1	Param_2	Param_3	Param_4
o1	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%s: Program Description

The current program description can be enquired using:

CMD	Param_1	Param_2
o1	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Bitstream Mode o2

This command sets the program bitstream mode to indicate the program service type:

CMD	Param_1	Param_2	Param_3	Param_4
o2	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02x: Bitstream mode 0 – Complete main 1 – Music and effects 2 – Visually impaired 3 – Hearing impaired 4 – Dialogue 5 – Commentary 6 – Emergency 7 – Voiceover 8 – Karaoke

The current program bitstream mode can be enquired using:

CMD	Param_1	Param_2
o2	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Audio Coding Mode o3

This command sets the program audio coding mode to indicate the main service channels in use:

CMD	Param_1	Param_2	Param_3	Param_4
o3	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%s: Audio Coding Mode 0 – Two independent channels 1 – Centre channel 2 – Left, right channels 3 –Left, centre, right channels 4 – Left, right, surround channels 5 –Left, Centre, Right, Surround channels 6 – Left, right, surround left, surround right channels 7 – Left, centre, right, surround left, surround Right channels

The current audio coding mode can be enquired using:

CMD	Param_1	Param_2
o3	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Centre Down-Mix Level o4

This command sets the program nominal down-mix level of the centre channel compared to the left and right channels:

CMD	Param_1	Param_2	Param_3	Param_4
o4	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Centre Mix Level 0 – -3dB 1 – -4.5dB 2 – -6dB

The current program centre down-mix level can be enquired using:

CMD	Param_1	Param_2
o4	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Surround Down-Mix Level o5

This command sets the program nominal down-mix level of the surround channels:

CMD	Param_1	Param_2	Param_3	Param_4
o5	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Surround Mix Level 0 – -3dB 1 – -6dB 2 – 0dB

The current program surround down-mix level can be enquired using:

CMD	Param_1	Param_2
o5	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Dolby Surround Mode o6

This command sets the program Dolby surround mode:

CMD	Param_1	Param_2	Param_3	Param_4
o6	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Dolby Surround Mode 0 – Not indicated 1 – Dolby surround not encoded 2 – Dolby surrounded encoded

The current program Dolby surround mode can be enquired using:

CMD	Param_1	Param_2
o6	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program LFE Enable o7

This command sets the program Low Frequency Effects (LFE) enable:

CMD	Param_1	Param_2	Param_3	Param_4
o7	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: LFE Enable 0 – Disabled 1 – Enabled

The current program LFE enable can be enquired using:

CMD	Param_1	Param_2
o7	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Dialogue Normalisation o8

This command sets the program dialogue normalisation value in order to compensate for dialogue of different volumes. The normalisation value is the mean level for the dialogue in a program relative to 0dB full-scale digital level:

CMD	Param_1	Param_2	Param_3	Param_4
o8	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Dialogue Normalisation (0 to 31)

The current program dialogue normalisation can be enquired using:

CMD	Param_1	Param_2
o8	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Production Mix Level o9

This command sets the program production mix level value:

CMD	Param_1	Param_2	Param_3	Param_4
o9	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Mix Level (0 to 31)

The current program production mix level can be enquired using:

CMD	Param_1	Param_2
o9	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Production Room Type oA

This command sets the program production room type used for the final audio mixing:

CMD	Param_1	Param_2	Param_3	Param_4
oA	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Room Type 0 – Not Indicated 1 – Large, X curve monitor 2 – Small, flat monitor

The current production room type can be enquired using:

CMD	Param_1	Param_2
oA	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Copyright Flag oB

This command sets the program copyright flag for the information contained in the bitstream:

CMD	Param_1	Param_2	Param_3	Param_4
oB	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: Copyright Flag 0 – Not copyright 1 – Copyright

The current program copyright flag can be enquired using:

CMD	Param_1	Param_2
oB	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Original Bitstream Format oC

This command sets whether the program is an original bitstream, or a copy of another bitstream:

CMD	Param_1	Param_2	Param_3	Param_4
oC	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: Bitstream Frag 0 – Not original bitsream 1 – Original bitstream

The current program original bitstream format can be enquired using:

CMD	Param_1	Param_2
oC	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Down-Mix Mode **oD**

This command sets the type of stereo program down-mix mode preferred by the master engineer. Note that this may be used, overridden or ignored:

CMD	Param_1	Param_2	Param_3	Param_4
oD	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Preferred Down-Mix Mode 0 – Not indicated 1 – Left total/Right total preferred 2 – Left only/Right only preferred

The current program down-mix mode can be enquired using:

CMD	Param_1	Param_2
oD	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Down-Mix Lt/Rt Centre Mix Level **oE**

This command sets the nominal program down-mix level of the centre channels with respect to the left and right channels on a left total/right total (Lt/Rt) down-mix:

CMD	Param_1	Param_2	Param_3	Param_4
oE	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Centre Mix Level 0 – +3dB 1 – +1.5dB 2 – 0dB 3 – –1.5dB 4 – –3dB 5 – –4.5dB 6 – –6dB 7 – –infinity dB

The current program down-mix Lt/Rt centre mix level can be enquired using:

CMD	Param_1	Param_2
oE	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Down-Mix Lt/Rt Surround Mix Level **oF**

This command sets the nominal program down-mix level of the surround channel with respect to the left and right channels on a left total/right total (Lt/Rt) down-mix:

CMD	Param_1	Param_2	Param_3	Param_4
oF	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Surround Mix Level 0 – -1.5dB 1 – -3dB 2 – -4.5dB 3 – -6dB 4 – -infinity dB

The current program down-mix Lt/Rt surround mix level can be enquired using:

CMD	Param_1	Param_2
oF	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Down-Mix Lo/Ro Centre Mix **oG**

This command sets the program down-mix level of the centre channels with respect to the left and right channels on a only/right only (Lo/Ro) down-mix:

CMD	Param_1	Param_2	Param_3	Param_4
oG	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Centre Mix Level 0 – -1.5dB 1 – -3dB 2 – -4.5dB 3 – -6dB 4 – -infinity dB

The current program down-mix Lo/Ro centre mix level can be enquired using:

CMD	Param_1	Param_2
oG	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Down-Mix Lo/Ro Surround Mix Level oH

This command sets the program down-mix level of the surround channels with respect to the left and right channels on a only/right only (Lo/Ro) down-mix:

CMD	Param_1	Param_2	Param_3	Param_4
o	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Surround Mix Level 0 – -1.5dB 1 – -3dB 2 – -4.5dB 3 – -6dB 4 – -infinity dB

The current program down-mix Lo/Ro surround mix level can be enquired using:

CMD	Param_1	Param_2
oH	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Dolby Surround EX™ Mode oI

This command sets whether the program has been executed in Dolby Surround EX™ mode. This command is only applicable to audio coding modes of 2/2 or 3/2:

CMD	Param_1	Param_2	Param_3	Param_4
ol	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Dolby Surround EX™ mode 0 – Not indicated 1 – Not encoded 2 – Encoded

The current program Dolby Surround EX™ mode can be enquired using:

CMD	Param_1	Param_2
ol	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Dolby Headphone Mode oJ

This command sets whether the program has been Dolby headphone encoded. This command is only applicable to audio coding modes of 2/0:

CMD	Param_1	Param_2	Param_3	Param_4
oJ	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Headphone Mode 0 – Not indicated 1 – Not encoded 2 – Encoded

The current program Dolby headphone mode can be enquired using:

CMD	Param_1	Param_2
oJ	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program A/D Converter Type oK

This command sets the program Analogue to Digital (A/D) converter type used to capture PCM audio:

CMD	Param_1	Param_2	Param_3	Param_4
oK	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: ADC Type 0 – Standard (or unknown) 1 – HCD

The current program A/D converter type can be enquired using:

CMD	Param_1	Param_2
oK	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program High Pass Filter Enable oL

This command sets the program DC blocking High Pass Filter (HPF) enable for the main input channels of the Dolby Digital encoder:

CMD	Param_1	Param_2	Param_3	Param_4
oL	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: HPF Enable 0 – Disabled 1 – Enabled

The current program HPF enable can be enquired using:

CMD	Param_1	Param_2
oL	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Low Pass Filter Enable oM

This command sets the program Low Pass Filter (LPF) enable for the main input channels of the Dolby Digital encoder:

CMD	Param_1	Param_2	Param_3	Param_4
oM	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: LPF Enable 0 – Disabled 1 – Enabled

The current program LPF enable can be enquired using:

CMD	Param_1	Param_2
oM	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program LFE LPF Enable oN

This command sets the program Low Frequency Effects (LFE) Low Pass Filter (LPF) enable for the LFE channel input of the Dolby Digital encoder:

CMD	Param_1	Param_2	Param_3	Param_4
oN	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: LFE LPF Enable 0 – Disabled 1 – Enabled

The current program LFE LPF enable can be enquired using:

CMD	Param_1	Param_2
oN	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Surround 90 Degree Phase-Shift

oO

This command sets the program surround channels 3dB attenuation enable:

CMD	Param_1	Param_2	Param_3	Param_4
oO	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: 90 Degrees Phase Shift Enable 0 – Disabled 1 – Enabled

The current program 90 degree phase shift can be enquired using:

CMD	Param_1	Param_2
oO	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Surround 3dB Attenuation

oP

This command sets the program surround channel 3dB attenuation enable.
This command is used to compensate surround levels prior to encoding:

CMD	Param_1	Param_2	Param_3	Param_4
oP	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: 3dB Attenuation Enable 0 – Disabled 1 – Enabled

The current program3dB attenuation can be enquired using:

CMD	Param_1	Param_2
oP	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program RF Pre-emphasis oQ

This command sets the program radio frequency (RF) pre-emphasis for RF modulated Dolby Digital bitstreams:

CMD	Param_1	Param_2	Param_3	Param_4
oQ	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1: RF Pre-emphasis Enable 0 – Disabled (default) 1 – Enabled

The current program RF pre-emphasis can be enquired using:

CMD	Param_1	Param_2
oQ	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Compression Profile/Word oR

This command sets the program RF compression profile/word. A flag is set to indicate whether the compression is a profile or a word, and sets that actual profile or word:

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
oR	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%1x: Format 0 – Profile 1 – Word	%02: Word (0 to 255) or Compression Profile

If Param_3 indicates that the compression is a word is selected, the actual word is a value between 0 and 255. If Param_3 indicates that the compression is a profile is selected, the options for Param_4 are:

Value	Description
0	None
1	Film-standard
2	Film-light
3	Music-standard

4	Music-light
5	Speech

The current program compression profile/word can be enquired using:

CMD	Param_1	Param_2
oR	%1x: Bank No.	%1x: Program No.

The response is in the form of the set command.

Set Program Dynamic Range Value oS

This command sets the program compression profile:

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
oS	%1x: Bank No.	%1x: Program No.	%02x: Word No. (0 to 7)	%02x: Flags 00	%02: Word (0 to 255) or Compression Profile

If Param_3 indicates that the compression is a word is selected, the actual word is a value between 0 and 255. If Param_3 indicates that the compression is a profile is selected, the options for Param_4 are as described for oR.

The current program dynamic range value can be enquired using:

CMD	Param_1	Param_2	Param_3
oS	%1x: Bank No.	%1x: Program No.	%02x: Word No. (0 to 7)

The response is in the form of the set command.

Load Metadata Preset **oT**

This command loads a metadata preset:

CMD	Param_1	Param_2
oT	%1x: Bank No.	%s: Metadata Preset Name

The current metadata preset can be enquired using:

CMD	Param_1
oT	%1x: Bank No.

The response is in the form of the set command.

Set Program Dynamic Range Type **oU**

This command sets the program dynamic range type:

CMD	Param_1	Param_2	Param_3	Param_4
oU	%1x: Bank No.	%1x: Program No.	%02x: Flags 00	%02: Format 0 – Profile 1 – Word

The current program dynamic range type can be enquired using:

CMD	Param_1
oU	%1x: Bank No.

The response is in the form of the set command.

Easytext Commands

Introduction

The Oxtel Series automation protocol supports control of the Easytext character generation software when the Easytext licence is present. Commands have been added to the Imagestore software to allow specification and formatting of on-screen text.

Summary

Easytext templates are built using the Textbuilder software application from the Media Conversion Software (MCS) suite. Templates consist of a number of ‘boxes’ each of which can either contain a piece of text in a single style, or a particular graphic.

The number, positions, parameters and styles of these boxes are defined in the template, along with some optional ‘static text’ – the text shown when the template is first loaded.

For most simple automated CG (Character Generation) applications, the automation must perform three separate tasks: It must load the template, and then subsequently download new text to be displayed in certain boxes. On completion of this, it can take the template to air using the keyer controls.

Various problems may be found as this is attempted:

- The loading of a template happens as a ‘background’ operation: The ‘ACK’ from Imagestore simply means the load request has been received correctly, not that it has completed.
- Because of this, new text should not be sent until the template has been loaded. It’s possible to determine this by using the ‘disk busy’ bit returned as part of the “Enquire loaded image status” (O) command - Alternatively, a simple delay of a couple of seconds will normally suffice.
- Once the template is loaded, new text can then be sent. The box numbers specified when the template was created must be known, to ensure that the new text is inserted in the correct places. Again, the rendering of the text into the template is comparatively slow, and it may be important to wait for the text to be rendered before taking the template to air. This must currently be handled with a delay, which will depend on the complexity of the template and the number of characters to be drawn.

The Z0 command is the basis for all text updates.

All Easytext commands take a ‘layer’ parameter, defining whether the midground (preview) or foreground (program) image is being modified.

In addition, many commands refer to a specific box on the template. Box numbers can range from 0 to 254 (0xFE). Box 255 (0xFF) is treated as a special case by some commands.

International Character Set Support

Easytext uses Unicode rather than ASCII to represent characters. Unlike ASCII, which is limited in its scope to common western characters, Unicode is a character-encoding standard that encapsulates all the major scripts used throughout the world. Unicode characters are typically represented with either 16 or 32-bit codes, rather than the 8 bits of ASCII.

To transfer Unicode data into Easytext, a format known as UTF-8 must be used, which encodes each Unicode code into a sequence of one or more 8-bit values. For common Latin characters there is a one-to-one mapping between UTF-8 and standard ASCII. The UTF-8 code for these characters is a single byte. Codes for other characters can vary in length between 2 and 4 bytes. In the case of the Easytext automation command ‘Update Text Field’, it is therefore valid to use:

```
remote_send( "Z0%01x%02x%x%s", Layer, BoxNo,  
            Render, "Hello World");
```

However when characters outside the standard 7-bit set are required (including special European and accented characters) the text must be converted to the UTF-8 format. Routines to perform this conversion can be found by searching on the Internet.

After the encoding to UTF-8, the byte-stream must be processed in the same way as filenames and titles, to replace protocol-specific control codes with alternative coding. See the section on loading and saving normal images by filename for details of this conversion.

More Sophisticated Control of Easytext

Additional commands allow automation systems to override parameters set in the template – changing font sizes and colours, and even altering text box positions on screen.

Note: In order for one or more updated parameter, such as the text or an image for a particular box, to be rendered the “Render Box” command Z3 has to be sent.

Due to the large number of parameters which can be altered on any one text box, the process of setting up these parameters is typically separate from the process of ‘rendering’ the text using the parameters. Because most commands will not automatically cause the text to be rendered, a separate ‘Render Box’ command is provided. This allows multiple parameters to be set for a variety of boxes, and then the individual box (or complete template) can be re-rendered.

Example

```
// Change box number 1 on layer 2 to Image.oxt
Z4201Image.oxt
// Render the box so the above change take effect
Z3201
```

Sending Colours via Automation

Various commands require colours to be sent to Easytext: All colours are sent as 6 bytes in RRGGBB format, with a range of 0 to 255 (0x00 to 0xFF) for each colour.

Examples:

Black	000000
White	FFFFFF
Red	FF0000
Green	00FF00
Blue	0000FF
Mid-grey	808080

Main Commands

Update Text Field **Z0**

This command updates the *text* in a specified text box. Formatting and style of the text is unchanged.

CMD	Param_1	Param_2	Param_3	Param_4
Z0	%01x Layer No.	%02x Text Box No.	%01x Render	%s String

Example:

```
void UpdateTextBox(int Layer, int BoxNo,
                  int flags, char* String)
{
    remote_send("Z0%01x%02x%01x%s", Layer, BoxNo,
                flags, String);
}
```

The layer value specifies which Imagestore keying layer¹⁰ is addressed. The Box number ranges from 0x00 to 0xFF and relates to which Text Box is to be updated.

The String is the new text to be written into text box specified. To enable international character support, the string should be UTF-8 encoded before transmission, see below.

Two bitwise flags are defined in the flags parameter:

- ET_RENDER (0x1)
- ET_APPEND (0x2)

If the ET_RENDER bit is set then the new text will appear on screen with all changes to font, size, colour and position that have been specified since the last call to Render Box (Z3) or Update Text Field (Z0). If ET_RENDER is not set then changes will not appear on screen until the next call to Render

¹⁰ In Swap-Preview mode layers 0 and 1 represent preview and program

Box (Z3) or Update Text Field (Z0) with ET_RENDER set. Imagestore Intuition does not use this bitwise flag so calls to Z0 will render immediately.

If the ET_APPEND bit is set, then the text is appended to the current text. This allows arbitrarily long text data to be defined over several command packets. Typically, the first packet should have flags = 0 (do not append, do not render), remaining packets should have ET_APPEND flag set, and the final packet should have ET_APPEND + ET_RENDER to display the completed text string.

Change Box Size and Position Z1

This command alters the position and dimensions of a text or image box.

CMD	Param_1	Param_2	Param_3	Param_4	Param_5	Param_6
Z1	%01x Layer No	%02x Box No.	%03x X Pos.	%03x Y Pos.	%03x Width	%03x Height

Example:

```
void ChangeDims(int Layer, int BoxNo, int X,
                int Y, int Width, int Height)
{
    remote_send("Z1%01x%02x%03x%03x%03x", Layer,
                BoxNo, X, Y, Width, Height);
}
```

The X and Y position indicate the top left position of the text box. All values are in pixels. This command can be used to alter the position of an image box.

This command will not update the text box on screen - the settings take effect when the 'Render Box' command is issued.

Render Box Z3

This command will update the specified text or image box on screen using all previous settings.

CMD	Param_1	Param_2
Z3	%01x Layer No.	%02x Text Box No.

If a box number of 255 (0xFF) is sent, then ALL boxes are re-rendered with their latest values. This is useful to guarantee that all boxes update simultaneously.

Note: This command must be sent to cause boxes to be re-rendered after adjusting their parameters via automation

Example:

```
void RenderBox(int Layer, int BoxNo)
{
    remote_send("Z3%01x%02x", Layer, BoxNo);
}
```

Change Image Z4

This command allows the image associated with an image box to be replaced with another image held on disk.

Only .OXT image files are supported for this command.

CMD	Param_1	Param_2	Param_3
Z4	%01x Layer No.	%02x Image Box No.	%s Image Filename

Example:

```
void UpdateBoxImage(int Layer, int BoxNo)
{
    remote_send("Z4%01x%02x%s", Layer, BoxNo,
                "NewLogo.oxt");
}
```

Set Text Font and Colour Z2

The font and colour of the text contained in a text box can be changed with this command.

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
Z2	%01x Layer No.	%02x Text Box No.	%02x Point Size	%06x Colour	%s Font File Name

Example:

```
void ChangeFont(int Layer, int BoxNo, int Size,
                int Col, char *FontFileName)
{
    remote_send("Z2%01x%02x%02x%06x%s", Layer,
                BoxNo, Size, Col, FontFileName);
}
```

The font name relates to the font filename to be used in the graphic. For example 'Times New Roman' must be addressed as 'times.ttf'.

The size is the 'point size' of the font. Note that points do not directly equate to pixels. The font's design determines this.

This command will not update the text box on screen - the settings take effect when the 'Render Box' command is issued.

Set Transparency ZA

The transparency of an image or text can be set with this command.

CMD	Param_1	Param_2	Param_3
ZA	%01x Layer No.	%02x Image / Text Box No.	%02x Transparency

Example:

```
void ChangeTextBgnd(int Layer, int BoxNo,
                     int Transparency)
{
    remote_send ("%01x%02x%02x", layer, BoxNo,
                Transparency);
}
```

A transparency of 255 (0xFF) is fully opaque, and 0 is transparent.

Set Text Drop Shadow **Z9**

The drop shadow of the text can be adjusted with this command.

CMD	Param_1	Param_2	Param_3	Param_4	Param_5	Param_6
Z9	%01x Layer No.	%02x Text Box No.	%02x X Offset	%02x Y Offset	%02x Trans.	%06x Colour

Example:

```
void ChangeTextBgnd(int Layer, int BoxNo,
                     int Xoff, int Yoff,
                     int Transparency, int Colour)
{
    remote_send ("%01x%02x%02x%02x%02x%06x", Layer,
                BoxNo, Xoff & 0xFF, Yoff & 0xFF,
                Transparency, Colour);
}
```

The X and Y offsets are the distance in pixels from the text which the drop shadow falls. Positive numbers are down and right. Do not forget to limit these numbers to two digits. Setting these both to zero turns off the drop shadow.

The transparency of the drop shadow can be set with ‘trans.’ where 0xFF is solid colour and 0x00 is transparent.

Set Text Tracking **ZB**

The tracking (internal inter-character spacing) of text can be modified with this command.

CMD	Param_1	Param_2	Param_3
ZB	%01x Layer No.	%02x Text Box No.	%04x Tracking

Positive numbers move the letters further apart, while negative tracking closes up the gaps between letters. The unit of tracking is 1/64th of a pixel.

The signed value should be ‘ANDED’ with 0xFFFF to ensure it is only four digits long.

Example:

```
void ChangeTextBgnd(int Layer, int BoxNo,
                     int Tracking)
{
    remote_send ("%01x%02x%04x", layer, BoxNo,
                 Tracking & 0xffff);
}
```

Set Text Alignment ZE

The alignment of text within a box can be adjusted with this command

CMD	Param_1	Param_2	Param_3	Param_4
ZE	%01x Layer No.	%02x Text Box No.	%01x Horiz. Align	%01x Vert. Align

Horizontal Align : 0 = Align Left, 1 = Align Centre, 2 = Align Right.

Vertical Align : 0 = Align Top, 1 = Align Centre, 2 = Align Bottom.

Set Text Wrapping ZF

The alignment of text within a box can be adjusted with this command

CMD	Param_1	Param_2	Param_3
ZF	%01x Layer No.	%02x Text Box No.	%01x Wrap Mode

A Wrap mode of 0 means no automatic wrapping occurs. If the text is too wide for the box, the point size is automatically reduced. Existing line breaks in the text are maintained.

If a wrap mode of 1 allows Easytext to insert additional line breaks between words to use the size of the text box without excessive shrinking of the text.

Text Box Background Settings

The following commands allow control of the backgrounds behind each text box.

Set Text Background to Clear Z8

A coloured or gradient background may be removed with this command.

CMD	Param_1	Param_2
Z8	%01x Layer No.	%02x Text Box No.

Example:

```
void TextBgnd_Clear(int Layer, int BoxNo)
{
    remote_send ("Z8%01x%02x%01x", layer, BoxNo);
}
```

Set Text Background to Matte Z6

This command allows a solid background behind a box to be automatically drawn (and resized) when the box is drawn

CMD	Param_1	Param_2	Param_4	Param_5	Param_6	Param_7
Z6	%01x Layer No.	%02x Box Number	%06x Colour	%02x Trans	%02x Horiz Border	%02x Vert Border

Example:

```
void TextBgnd_Matte(int Layer, int BoxNo, int Col,
                     int trans, int hborder,
                     int vborder)
{
    remote_send ("Z6%01x%02x%06x%02x%02x%02x",
                 layer, BoxNo, Col, trans,
                 hborder, vborder);
}
```

The transparency of the colour can be set with ‘trans’ where 0xFF is solid colour and 0x00 is transparent.

The horizontal border and vertical border parameters describe the horizontal and vertical border placed around the colour block. Border values of 0 will produce a rectangle which fits behind the rendered text exactly. Values of 10 (0x0A) would produce a 10 pixel border outside the text rectangle.

This command will take effect only when the text box is re-rendered.

Set Text Background to Gradient **Z7**

This command allows a gradient background behind a box to be automatically drawn (and resized) when the box is drawn

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
Z7	%01x Layer No.	%02x Box No.	%06x Source Colour	%06x Dest. Colour	%06x Border Colour
Param_6	Param_7	Param_8	Param_9	Param_10	
	%02x Trans.	%02x Border Width	%03x Direction	%02x Horiz. Border	%02x Vert. Border

Example:

```
void TextBgnd_Gradient(int Layer, int BoxNo,
                        int SourceCol, int DestCol,
                        int BorderCol, int Trans,
                        int BorderWidth, int Dir,
                        int hborder, int vborder)
{
    remote_send("Z7%01x%02x%06x%06x%06x%02x%02x%03x"
                "%02x%02x", Layer, BoxNo, SourceCol,
                DestCol, BorderCol, Trans,
                BorderWidth, Dir, HBorder, VBorder);
}
```

The gradient starts from the source colour, and finishes at the destination colour.

The border colour is drawn around the gradient-filled text box.

The transparency of the box background can be set with ‘trans’ where 0xFF is solid colour and 0x00 is transparent.

If the direction variable is 0, the gradient will be vertical; otherwise the gradient will be horizontal.

The hborder and vborder define the width of the horizontal and vertical borders placed around the gradient colour block. Border values of 0 will produce a rectangle which fits behind the rendered text exactly, values of 10 (0x0A) will produce a 10 pixel border around the text rectangle. This command will take effect only when the text box is re-rendered.

Setting Template background options ZD

As well as specifying backgrounds for individual boxes, it is possible to specify a background for the entire template. The background size will be the size of the template created in Text Builder.

Template backgrounds can be a solid colour, with controllable transparency, or can have transparency that fades in and/or out.

CMD	Param_1	Param_2	Param_3	Param_4	Param_5
ZD	%01x Layer No.	%06x Colour	%02x Transp	%01x Mode	%02x Edgesize

The mode values can be:

0	Solid
1	Fade Down
2	Fade Up
3	Fade Both
4	Fade Edge

For mode 4, the Edge Size parameter determines the width of the faded area.

Example:

```
void ChangeTextBgnd(int Layer, int Colour,
                     int Trans, int mode,
                     int EdgeSize)
{
    remote_send ("ZD%01x%06x%02x%01x%02x", layer,
                 Colour, Trans, mode, EdgeSize);
}
```

Text Straps

Easytext supports text rolls and crawls, known together as straps. Additional commands give extra control of the playout of these.

It is important to note that because the transition between ‘old’ and ‘new’ text happens at the wrapping point of the strap, there may be a significant delay between sending the text message, and having it displayed. There is no ‘queue’ for messages: If new text is sent before previously transmitted text has been displayed, the previously transmitted text may never be seen.

Run Strap **Z5**

This command allows an Easytext Strap to be stopped and re-started. If an Easytext strap is stopped and re-started it will start from its original location rather than from where it was stopped.

CMD	Param_1	Param_2
Z5	%01x Layer No.	%01x Run Flag 0 = Stop, 1 = Run

Example:

```
void StartStrap(int Layer, bool Start)
{
    remote_send("Z5%01x%01x", Layer, Start);
}
```

Set Strap Speed **ZC**

This command allows the speed of an Easytext Strap to be controlled

CMD	Param_1	Param_2
ZC	%01x Layer No.	%02x Speed : pixels/lines per field

Only positive and even speeds (i.e. 0, 2, 4, 6, 8, etc) are supported.

Enquire Crawl Count Loop

Zc

This command returns information about how many times a Crawl has looped.

The loop count will be set to zero when a crawl is first loaded, or when the crawl is cut/faded up.

CMD	Param_1
Zc	%01x Layer No.

Example:

```
void Enq_Crawl_Count_Loop(int Layer)
{
    remote_send("Zc01x", Layer);
}
```

Imagestore Intuition requires an extra parameter “Box number” because there can be more than one crawl loaded into each layer.

CMD	Param_1	Param_2
Zc	%01x: Layer No.	%02x Box No.

Example:

```
void Enq_Crawl_Count_Loop(int Layer, int Box_no)
{
    remote_send("Zc%01x %02x", Layer, Box_no);
}
```

The information returned is:

Format	Field	Description
%c%c	Cmd	Zc
%4d	Loop Count	How may times the crawl has cycled around

Text Box Update **Zd**

This command sends the Imagestore a remote Easytext command to update a text box within a specified loaded template.

The command format is:

CMD	Param_1	Param_2	Param_3	Param_4	Param_5	Param_6
Zd	%01x Layer No.	%02x Box No.	%01x Flags	%02x Filename string length	%s Filename	%s Text

Parameter	Description
Layer No.	See page 269 for details
Box No.	Text box number
Flags	Render flags
Filename string length	The string length of the filename of the file to be address
Filename	Filename
Text	Text to be rendered

Example:

```
"Zd10010dTransTest.oxaNumbers 123456789"

Zd = Command
1 = Layer (DSK 2)
00 = Text Box Number
1 = Flags
0d = String Length of Filename we want to
      write to (strlen(TransTest.oxa))
TransTest.oxa = File we want to address
"Numbers 123456789" The text we need to render
```

This results in the string "Numbers 123456789" being rendered in the template "TestTrans.oxa" if it were loaded into DSK2.

Image Update Ze

This command sends the Imagestore a remote Easytext command to update an image within a loaded template. If a non-existent filename is supplied as Param_6, no action is taken.

The command format is:

CMD	Param_1	Param_2	Param_3	Param_5	Param_6
Ze	%01x Layer No.	%02x Box No.	%02x Filename string length	%s Filename	%s Image

Parameter	Description
Layer No.	See page 269 for details
Box No.	Image box number
Filename string length	The string length of the filename of the file to be address
Filename	Filename
Text	Image to be added

Example:

```
"Ze1010dTransTest.oxaAdvert2.oxt"
```

Would update an image in the template TransTest.oxa with the image Advert2.oxt

The image command uses the prefix 'Ze' where the text equivalent uses the prefix 'Zd'.

Enquire Textbox Za

Enquires information about a particular box within an Easytext template.

CMD	Param_1	Param_2
Za	%01x Layer No.	%02x Box No.

If a valid text box is found it returns with:

Format	Field	Description
%c%c	Cmd	Za
%03x	X Position	In pixels
%03x	Y Position	In pixels
%03x	Width	In pixels
%03x	Height	In pixels
%02x	Font Point Size	
%06x	Box Colour	RGB
%02x	Transparency	0 (Transparent) -255 (0xFF) (Solid)
%04x	Tracking	Font Tracking
%01x	Horizontal Alignment	LEFT =0, CENTRE=1, RIGHT=2
%01x	Vertical Alignment	TOP=0, CENTRE=1, BOTTOM=2
%02x	Drop Shadow X offset	In pixels
%02x	Drop Shadow Y offset	In pixels
%02x	Drop Shadow Transparency	0 (Transparent) -255 (0xFF) (Solid)
%06x	Drop Shadow Colour	RGB colour
%01x	Word Wrap	1= on, 0= off
%s	Font Name	

Enquire Template **Zb**

Returns information about the currently loaded template.

CMD	Param_1
Zb	%01x Layer No,

If a template is found in that layer, the information returned is:

Format	Field	Description
%c%c	Cmd	Zb
%02x	Number of Objects	Number of text + image boxes
%06x	Background Colour	RGB
%02x	Transparency	0 (Transparent) -255 (Solid)
%01x	Background Gradient	SOLID=0, DOWN=1, UP=2, CENTER=3, EDGE=4
%02x	Edge	Size of edge in pixels (If Gradient is edge).

Stop Animation **Zf**

On Intuition units, animations occupy a template box rather than a full layer. The Zf command behaves exactly as the S1 command except that it includes a box number.

CMD	Param_1	Param_2	Param_3
Zf	%01x Layer No.	%02x Text Box No.	%01x Complete Cycle Flag 0 = Complete cycle 1 = Stop immediately 2 = Complete cycle and restart (in-loop- out animations only)

Pause/Restart Animation **Zg**

This command is used to pause or restart a roll or crawl. A value of 1 pauses the roll/crawl, and 0 will restart it. The box is addressed by layer on Imagestores as follows:

CMD	Param_1	Param_2
Zg	%01x Layer No.	%01x Pause = 1, Restart = 0

Imagestore Intuition requires an extra parameter “Box number” because there can be more than one crawl loaded into each layer

CMD	Param_1	Param_2	Param_3
Zg	%01x Layer No.	%02x Box No.	%01x Pause = 1, Restart = 0

Set Text Crawl Count **Zh**

This command allows the user to specify the text crawl count for a box on a layer, thereby overriding the default ‘Loop Count’ value that is set within the Intuition template file (or overriding ‘Continuous’ looping). The contents of the Intuition template file will not be updated as a result of the Zh command.

The crawl count can be sent dynamically whilst the crawl is already in progress. If the new crawl count is less than the actual count, then the text crawl will end.

The command format is:

CMD	Param_1	Param_2	Param_3	Param_4
Zh	%01x Layer No	%02x Box No.	%04x New Crawl Count	%01x: Auto cut down 1 = Strap should cut down after reaching maximum number of loops. 0 = Strap should not cut down

Datasources

Datasources may be used to simplify delivery of textual information into Easytext-equipped systems. A datasource is a named tag embedded within the string of an Easytext box. Datasources can be created, modified or deleted under automation control. By updating the string value of the datasource dynamically, the string can be modified while it is on-air (either static or scrolling).

In addition to external datasources, there are assorted internal datasources that can be accessed by templates. These include time, date and other system information.

Within an Easytext message, a datasource reference is encoded as follows:

```
|datasourcename[:option=value]|
```

The vertical bars delimit the datasource reference. The ':option=value' section may be omitted, or may have multiple instances, adding different options

Examples:

```
|TIME| could produce "11:55"  
|TIME:FORMAT "H:M:S" | might produce "11:55:01"  
|TIME:FORMAT "H:M":OFFSET "-1" | could produce  
"10:55" (multiple options, each of which could be  
omitted)
```

Options (for internal datasources only) are specified after a colon. The option name is followed by a quoted value string. The interpretation of options and their values is entirely dependent on the internal datasource.

Naturally, this means that vertical bar characters cannot be easily encoded. A sequence of two vertical bars will generate a single vertical bar, so vertical bars can be inserted using the sequence 'll'

All datasource names, option names and values are case-sensitive and are stored as Unicode values.

The advantage of using datasources with Easytext is that the process of delivering textual data can become entirely separated from the process of recalling slides that refer to it.

For example, it is possible to have one PC feeding in a news headline summary into a datasource called 'NEWS', while another PC feeds stock ticker info into a datasource called 'STOCKS'. If we have two templates, one referring to 'STOCKS' and the other 'NEWS', then depending on which template is loaded, the appropriate information will be displayed. This happens regardless of how the template is loaded, and the information-providing PCs are unaware of what is happening 'on-air'. Also, the process of issuing new templates becomes easier, as the requirement for automation to address particular numbered text boxes is removed.

Another advantage of datasources is that datasources can be inserted anywhere into a text box, allowing composite messages to be generated using only a single text box: For example: "The time is |TIME| on |DAYOFWEEK|"

Datasources created by network automation sessions remain valid when the session is disconnected, but datasource values are volatile: All values set under automation are lost when the unit restarts

All internal (pre-defined) datasources use upper-case names.

Datasource Automation Commands

Set Global Datasource Value **m0**

This command assigns a new value to an external datasource. The format of the command is as follows:

CMD	Param_1	Param_2
m0	%02x: Flags	%s: Name Value

The ‘Flags’ value gives a means of sending long strings by concatenation.

Bit 0 = 1: Set for the FIRST packet, clear for all others

Bit 1 = 1: Set for the LAST packet, clear for all others.

The ‘Name’ and ‘Value’ parameters consists of the datasource name and its new string value, separated by a vertical bar ‘|’. Vertical bars, colons and semi-colons may not appear within string values unless escaped.

Example1:

```
"m001MySource|This is the start"  
"m000MySource|, and this is some more"  
"m002MySource|, but this is the end!"
```

This gives MySource = “This is the start, and this is some more, but this is the end!”

Example2:

```
"m003MySource|This is the lot."
```

The datasource will not be used on-air until all packets have been received. Scrolling text will not update with a new datasource until the text box starts its next pass.

A global datasource can be used by any layer. Note that attempting to access a missing datasource will return an empty string.

Delete Global Datasource m1

This command deletes a global datasource, freeing up any memory used for it.

CMD	Param_1
m1	%s: Datasource Name

Set Layer-Specific Datasource Value m2

This command assigns a new value to an external datasource associated with a specific layer. The format of the command is as follows:

CMD	Param_1	Param_2	Param_3
m2	%02x: Flags	%02x: Layer No.	%s: Name Value

The ‘Flags’ value gives a means of sending long strings by concatenation. The format of the ‘Flags’, ‘Name’ and ‘Value’ parameters are as described for the m0 command.

The datasource is defined by the layer number and the name, and can only be used by the layer for which it was set. Note that attempting to access a missing datasource will return an empty string.

Delete Layer-Specific Datasource m3

This command deletes a datasource associated with a specific layer, freeing up any memory used for it.

CMD	Param_1	Param_2
m3	%1x: Layer No.	%s: Name

List Global Datasources m4

This command lists the names of all the global datasources available for use.

CMD
m4

The information returned is a list of the names of all the global datasources available, separated by a ‘|’ . If datasources ‘Name_1’ to ‘Name_n’ are set the response will be:

Format	Field	Description
%c%c	Cmd	m4
%s	Name_1 Name_2Name_n	List of global datasource names, or a blank string if there are no global datasources

The global datasources can be set using the m0 command, and deleted using the m1 command.

Query Global Datasource m5

This command shows the contents of a global datasource.

CMD	Param 1
m5	%s: Name

The information returned for a valid global datasource is:

Format	Field	Description
%c%c	Cmd	m5
%s	Name Value	Name of global datasource enquired, and the contents of that datasource or an empty string

Attempting to access a missing datasource, or a layer-specific datasource will return an empty string, and the ‘Value’ of the above response is empty.

List Layer-Specific Datasources m6

This command lists the name of all datasources associated with a specific layer.

CMD	Param_1
m6	%1x: Layer No.

The information returned is a list of the names of all the global datasources available, separated by a ‘|’ . If datasources ‘Name_1’ to ‘Name_n’ are set the response will be:

Format	Field	Description
%c%c	Cmd	m4
%1x	Layer No.	Layer number
%s	Name_1 Name_2Name_n	List of datasource names for that layer, or a blank string if there are no datasources for that layer

The layer-specific datasources can be set using the m2 command, and deleted using the m3 command.

Query Layer-Specific Datasource **m7**

This command shows the contents of a layer-specific datasource.

CMD	Param_1	Param 1
m7	%1x: Layer No.	%s: Name

The information returned for a valid layer-specific datasource is:

Format	Field	Description
%c%c	Cmd	m7
%1x	Layer No.	Layer number
%s	Name Value	Name of global datasource enquired, and the contents of that datasource or an empty string

Attempting to access a missing datasource, or a global datasource will return an empty string, and the ‘Value’ of the above response is empty.

Delete All Datasources **m8**

This command deletes all datasources (global and layer-specific), freeing up any memory used for them.

CMD	Param_1
m8	%1x: 0

Internal Datasources

Internal datasources cannot be overridden or deleted: This is a list of internal datasources in the version of software, along with any parameters

TIME

Returns the current time of day.

The option 'FORMAT' determines the format. All characters in the format are copied to the output, except the following ones, which are interpreted to insert time information

- i Insert hours - 12 hour clock, leading zeroes.
- h Insert hours - 12 hour clock, no leading zeroes.
- H Insert hours - 24 hour clock, leading zeroes.
- M Insert minutes - 2 digits, leading zeroes.
- S Insert seconds - 2 digits, leading zeroes. Use of this field will only work in very small, simple templates.
- A Insert 'AM' or 'PM' accordingly.
- a Insert 'am' or 'pm' accordingly.

Examples:

```
| TIME:FORMAT="h:M" |      // "09:15"  
| TIME:FORMAT="i:M.S" |      // "9:15.42"  
                           // 'S' not recommended
```

The OFFSET option allows time-zone adjustments. The offset is specified in seconds.

```
| TIME:OFFSET= -3600 |      // "08:15"  
| TIME:OFFSET= 3600  |      // "10:15"
```

DATE

The option ‘FORMAT’ determines the format. All characters in the format are copied to the output, except the following ones which are interpreted to insert date information

- D Insert day of the month - leading zeroes
- d Insert day of the month - no leading zeroes
- M Insert month number - leading zeroes
- m Insert month number – no leading zeroes
- Y Insert year number – 4 digits
- y Insert year number – last 2 digits
- W Insert full name of day of week (e.g ‘Monday’)
- n Insert day number suffix ‘st’, ‘nd’, ‘rd’ or ‘th’ as appropriate
- O Insert Month name ‘November’
- o Insert short Month name ‘Nov’

Examples:

```
| DATE : FORMAT = "D/M/Y" | // "01/04/2002"
| DATE : FORMAT = "W n O" | // "Sunday 2nd December"
```

An OFFSET parameter (also in seconds, like the TIME command), allows day rollover to be synchronized for other time zones.

FADER0 through FADER7

Returns the value of an analogue input fader, modified according to the options.

Analogue faders return a nominal value between 0 and 1. The voltages representing 0 and 1 must be discovered empirically.

‘SCALE’ Scaling factor for the value – defaults to 1.0

‘OFFSET’ Offset value for the number – defaults to 0.0

The value shown will be (FADERVALUE * SCALE) + OFFSET

‘DECIMALS’ determines how many decimal places are displayed with the number. This value defaults to 2.

HOSTNAME

Returns the hostname of the unit.

IP_ADDR

Returns the Ethernet IP address of the unit.

SERIAL

Returns the serial number of the unit.

SYSNAME

Returns a brief description of the software and version

MPEG Commands

Note: The following commands only apply to the Imagestore 3 with the MPEG hardware decoder option.

MPEG Control v0

The v0 command is used to control the MPEG Play-out card in Imagestore 3.

CMD	Param_1
v0	%02x

Where:

CMD	Description
v000	Stop MPEG Clip
v001	Play MPEG Clip
v002	Pause MPEG Clip
v003	Rewind MPEG Clip
v004	Next MPEG Clip
v005	Previous MPEG Clip
v006	Go to position HH MM SS FF (Hours minutes seconds frames with spaces)

MPEG Load v1

This command loads a MPEG file WITHOUT invoking a keyer.

CMD	Param_1
V1	%s Filename

This command plays the relevant MPEG clip which then appears on the SDI output of the IS3 without affecting what else is loaded into the keyers, in this way IS3 could be configured as a clip server.

EAS Commands

EAS On n1

Brings to air in the Foreground/Program keyer an EAS crawl with the specified priority (see command n4) and with the specified text (see command n3).

CMD
n1

Example:

n1

EAS Off n2

Whilst an EAS crawl is on air, this command restores the previously loaded image back to the Foreground/Program keying layer.

CMD
n2

Example:

n2

EAS Set Text n3

This command is used to set the text that will be displayed in the next EAS crawl.

CMD	Param_1
n3	%s Text

Example:

n3This is an EAS crawl

The text "This is an EAS crawl" is saved to the EAS crawl file specified by the current priority. The next time that the EAS On (n1) command is received, this text will be displayed.

Note: After using the 'n3' command, it is recommended to wait for three (3) seconds before issuing the EAS On (n1) command. This is to allow the Imagestore to store the text that has been set.

EAS Set Priority n4

This command is used to set the priority of the next EAS crawl, which will be displayed on air.

CMD	Param_1
n4	%02d Priority

Examples:

```
n401      // Sets to the highest priority (1)
n402      // Sets to the medium priority (2)
n403      // Sets to the lowest priority (3)
```

Note: After using the 'n4' command, it is recommended to wait 3 seconds before issuing the EAS On (n1) command. This is to allow the Imagestore to store the priority that has been set.

Passive Mode Commands

Passive mode is a mode of operation where the Imagestore will transmit its internal state via the automation port when something has changed. In this mode, the amount of polling required by external automation devices will reduce, as it enables them to listen to the Imagestore tallies to always have an accurate image of the system status.

Set Passive Mode On/Off **Ya**

This command switches passive mode on or off within the Imagestore.

CMD	Param_1
Ya	%01x 0=mode off, 1=mode on

Set Passive Mode Transmission Interval **Yb**

This command determines the interval (in milliseconds) at which the Imagestore 2/3 will check its internal state before relaying information to the outside world via automation.

CMD	Param_1
Yb	%04x Interval duration (ms)

The default value is 500ms (1/2 second, 0x1F4). Once this value has been changed it will be saved in the Imagestore state file for subsequent use.

Note: This command is only valid for Imagestore 2/3, or for Imagestore 300/HD-TV operating with an Easysound standalone. Later Imagestore products do not support this command because they are able to transmit state tallies every field/frame.

Layer Rate Yc

This tally is issued by the Imagestore 2/3 in passive mode whenever one of the layer fade or fade-to-black rates changes on the Imagestore. The format of the Imagestore 2/3 tally is as follows:

CMD	Param_1	Param_2	Param_3	Param_4
Yc	%03x Midground rate	%03x Foreground rate	%03x Midground FTB rate	%03x Foreground FTB rate

Strap Loop Count Yd

This tally is issued by the Imagestore 2/3 in passive mode whenever one of the strap loop counts increments on the Imagestore. The format of the Imagestore 2/3 tally is as follows:

CMD	Param_1	Param_2
Yd	%08x Midground loop count	%08x Foreground loop count

Enquire Passive Mode Ye

This command allows a remote device to check if Passive mode is on or off.

CMD
Ye

The response is:

- ‘Ye00’ if Passive mode is currently off
- ‘Ye01’ if Passive mode is currently on

Scheduled Commands

This feature allows the user to specify the time and frame number an automation command is to be executed.

Add Scheduled Command i0

This command schedules an automation command at a specified time up to 23 hours in the future.

CMD	Param_1	Param_2
i0	%08d Timecode	%s Automation command

A typical Imagestore automation command is in the form:

STX <Command> : CRC

Whereas the scheduled command is;

STX i0 <Timecode> ; <Command> : CRC

For multiple commands use a concatenated list:

STX i0 <Timecode> ; <Command1> ; <Command2> ;
<CommandN> : CRC

Note: Spaces shown in the command line are for clarity only and should not be used.

Timecode

The time is referenced to the currently selected timecode source, VITC or LTC and is formatted as an 8 digit number HHMMSSFF, where:

HH	= Hours	00 – 23
MM	= Minutes	00 – 59
SS	= Seconds	00 – 59
FF	= Frames	00 – 29

Commands can be scheduled for any time up to 23 hours in advance and midnight crossing is supported.

Note: If no timecode source is present the scheduler will not operate correctly.

Delete Scheduled Command i1

This command deletes a command that was scheduled at the specified time.

CMD	Param_1
i1	%08d Timecode

Example:

```
i1HHMMSSFF // Will delete all scheduled  
// commands at the specified timecode
```

Delete All Scheduled Command i2

This command deletes all scheduled commands.

CMD
i2

Time Of/Till Next Event i3

This command enquires the time of the next command, or the time until the next command

CMD	Param_1
i3	%01d: Flag 0=Timecode of next scheduled event, 1= Time until the next scheduled event

The response is:

CMD	Param_1	Param_2
i1	%01d Flag	%08d Timecode

Scheduled commands are NOT saved therefore powering off the Imagestore will clear the scheduled list.

If the video standard is changed, the scheduled list will be cleared.

Care must be taken when manual control of the Imagestore is undertaken as scheduled events will continue that may no longer be required. It is recommended that the i2 command be used to clear the scheduled list before operating the unit manually.

IntuitionXG Commands

Note: The following commands only apply to the IntuitionXG. Note that a space separates the command and the first parameter. The “|” character is used to separate each parameter from the next parameter in the command.

Cue V0

This sets a template to be cued onto a IntuitionXG keyer, which can be identified using the keyer number, or the keyer name that is used in Xplay.

CMD	Param_1	Param_2	Param_3
V0	%s Keyer Number or Keyer Name in Xplay	%x Template ID Type 0 – Recall ID 1 – Name 2 – Embedded Page	%s Template ID

If the template ID is specified as ‘Name’, the name used as Param_3 can be specified as the name of the page, template, panel or scene itself without any category information (the ‘short’ name), or it can be specified as a fully qualified name. Note that since ‘|’ is a reserved character, a fully qualified name would have to be written using escaped characters.

Example:

To cue keyer ‘1’ with an page "XMS:Pages.Demo:PG_Lower3rd":

V0 1 1 XMS\3APages.Demo\3APG_Lower3rd

If the template ID is specified ‘Embedded Page’, the embedded page xml must be specified. This will contain characters that need to be escaped.

Take V1

This causes items cued on IntuitionXG keyers to be taken. The keyers can be identified using the keyer number, or the keyer name used in Xplay. Multiple keyers are separated by the ‘,’ character and can be taken using one command.

CMD	Param_1	Param_2 - Optional
V1	<p>%s, %s, ...</p> <p>Keyer Numbers or Keyer Names in Xplay, separated by commas</p>	<p> %x</p> <p>Take Number</p> <p>-1 – Do all takes (default)</p> <p>0 – do first take</p> <p>1 – do second take ...</p>

Clear**V2**

This causes items that are on air on IntuitionXG keyers to be cleared. The keyers can be identified using the keyer number, or the keyer name used in Xplay. Multiple keyers are separated by the ‘,’ character and can be cleared using one command.

CMD	Param_1
V2	<p>%s, %s, ...</p> <p>Keyer Numbers or Keyer Names in Xplay, separated by commas</p>

Keyer Cut/Fade Up V3

This causes IntuitionXG keyers to be cut or faded up. The keyers can be identified using the keyer number, or the keyer name used in Xplay. Multiple keyers are separated by the ‘,’ character and can be cut/faded up using one command. The fade rate is specified in mS, where any positive integer. A fade rate of 0 indicates a cut.

CMD	Param_1	Param_2	Param_3 - Optional
V3	<p>%s, %s, ...</p> <p>Keyer Numbers or Keyer Names in Xplay, separated by commas</p>	<p> %x</p> <p>Fade Rate</p> <p>0 – Cut</p> <p>Fade rate in mS</p>	<p> %x</p> <p>Output</p> <p>0 – Preview</p> <p>1 – On air</p>

Keyer Cut/Fade Down V4

This causes IntuitionXG keyers to be cut or faded down. The keyers can be identified using the keyer number, or the keyer name used in Xplay. Multiple keyers are separated by the ‘,’ character and can be cut/faded down using one command. The fade rate is specified in mS, where any positive integer. A fade rate of 0 indicates a cut. The ‘Output’ parameter is optional.

CMD	Param_1	Param_2	Param_3 - Optional
V4	%s, %s, ... Keyer Numbers or Keyer Names in Xplay, separated by commas	%x Fade Rate 0 – Cut Fade Rate in mS	%x Output 0 – Preview 1 – On air

Set Property V5

This allows properties to be set on IntuitionXG. For example the text or image to be used within one or more template on one or more keyer can be set. The property can be set to be changed on the preview or on air. The default is for the property to change to be seen on air.

CMD	Param_1	Param_2	Param_3	Param_4 - Optional
V5	%s, %s, ... Objects and Properties	%s, %s, ... Values	%s, %s, ... Keyer Numbers or Keyer Names in Xplay.	%x Preview or On Air 0 – Preview 1 – On Air (default)

Multiple properties can be set by separating each ‘Objects and Properties’, ‘Values’ with the ‘,’ character. The ‘Objects and Properties’ to be set are a delimited list in the form “ObjectName.PropertyName”. The ‘Values’ indicate, for example, the text or image to be used for that object and name. The keyers can be identified using the keyer number, or the keyer name used in Xplay.

Example

To set

Object:	Textbox1
Property:	Text
Value:	Hello World!

And:

Object:	Image1
Property:	ImageName
Value:	XMS:Images.Headshots:Obama

On:

Keyers:	1, 2
Preview or On Air:	On Air

V5	Textbox1.Text, Image1.ImageName	Hello World!,
XMS\3AImages.Headshots\3AOObama		1, 2 1

Enable IntuitionXG Tallies V8

This enables or disables IntuitionXG tallies.

CMD	Param_1
V8	%d Enable/Disable 0 – Disable Tallies 1 – Enable Tallies

If 'V8 1' is sent to enable IntuitionXG tallies, then the tallies returned from Xplay will differ from those returned when Y61 is used (to enable video tallies) as follows:

- The scene event summary will report, for each keyer, the keyer state (3), the item that is cued (Y9) and the item that is on-air (V1)
- A V1 tally will be reported when an item is taken to air
- A V2 tally will be reported when an item is cleared from air

Y60 and Y61 still work as before.

The V1 and V2 tallies have the following formats:

Format	Field	Description
%c%c	Cmd	V1
%x	Keyer No.	Number of the keyer that was taken
%s	Item	Name of the item that was taken to air

Format	Field	Description
%c%c	Cmd	V2
%x	Keyer No.	Number of the keyer that was cleared

Miscellaneous Commands

Image Count **Ra**

This command returns how many images are used on the Imagestore 2/3 system and the maximum available.

CMD
Ra

The response is in the form:

CMD	Param_1	Param_2
Ra	%04x Number of images currently used	%04x Maximum number of images

Pixel Count **Rb**

Command 'Rb' responds with the total number of pixels in an animation on an Imagestore 2/3 or Imagestore HD-TV. This will allow automation vendors to dynamically determine the 'pre-roll' time needed when loading animations.

CMD	Param_1
Rb	%s Filename

The response is in the form:

CMD	Param_1
Rb	%x Total number of pixels. 0 Indicates that the file does not exist or a problem was encountered.

As an approximate guide Imagestore will load 30,000 pixels per second.

Appendix A

Command Validity

In the following please note that a number indicate the version number in which the command was introduced.

Command	Imagesstore 2/3	Imagesstore 300[+]/HD-TV	Intuition[+]	IntuitionXG	Imagesstore 750	DSK-3901	LGK-3901
0	X	X	X		1.14.1	2.1	2.1
1	X	X	X	4.0	1.14.1	2.1	2.1
2	X	X	X		1.14.1	2.1	2.1
3	X	X	X	4.0	1.14.1	2.1	2.1
4	X	X			1.14.1	2.1	2.1
5	X	X			1.14.1	2.1	2.1
6	X						
7	X						
8	X	X	X	4.0	1.14.1	2.1	2.1
9	X	X	X		1.14.1	2.1	2.1
@		X	X		1.14.1	2.1	2.1
A	X	X	X	4.2	1.14.1		2.1
B	X	X	X	4.0	1.14.1	2.1	2.1
C	Xa	X			1.14.1	2.1	2.1
D	X	X			1.14.1	2.1	2.1
E	X	X			1.14.1	2.1	2.1
F	X	X	X		1.14.1	2.1	2.1
G	X	X	X		1.14.1		2.1
H	X	X			1.14.1		2.1
I	X	X			1.14.1	2.1	2.1
J	X						
K	X	X			1.14.1	2.1	2.1
L	X	X			1.14.1	2.1	2.1
M	X	X	X	4.0	1.14.1	2.1	2.1
N	X	X	X	4.2	1.14.1	2.1	2.1

Command	LGK-3901	DSK-3901	Imagestore 750	IntuitionXG	Intuition[+]	Imagestore 300[+]HD-TV	2/3
O	X	X	X	4.0	1.14.1		2.1
P	X	X			1.14.1		2.1
Q	X	X	X		1.14.1		2.1
a	X						
b	X						
c	X						
d	X						
e	X						
f	X						
g	X						
h	X						
I					1.14.1		
m0		X	X	4.2	1.14.1		2.1
m1		X	X		1.14.1		2.1
m2		1.13.8	1.13.8	4.2	2.0.1		2.1
m3		1.13.8	1.13.8		2.0.1		2.1
m4		1.13.8	1.13.8		2.0.1		2.1
m5		1.13.8	1.13.8		2.0.1		2.1
m6		1.13.8	1.13.8		2.0.1		2.1
m7		1.13.8	1.13.8		2.0.1		2.1
m8		1.13.8	1.13.8		3.0		2.1
R0	X	X	X	4.0	1.14.1		2.1
R1	X	X	X		1.14.1		2.1
R2	X	X	X		1.14.1		2.1
R3	X	X	X	4.2	1.14.1		2.1
R4	X	X	X	4.0	1.14.1		2.1
R5	X	X	X	4.0	1.14.1		2.1
R6	X	X	X	4.0	1.14.1		2.1
R7	X	X			1.14.1		2.1
R8	X	X			1.14.1		
R9		X			1.14.1		2.1

Command	LGK-3901
Rm	X
RA	4.0
RB	2.27
	1.13.8
S0	X
S1	X
S2	X
S3	X
S4	X
T0	X
T1	X
T2	X
T3	X
U0	X
U1	X
U2	X
U3	X
U4	X
U5	X
U6	X
U7	X
U8	X
U9	X
UA	X
UB	X
UC	X
UD	X
UE	Xk
UF	
Ua	X
Ub	X

Command	LGK-3901	DSK-3901	Imagestore 750	IntuitionXG	Intuition[+]	Imagestore 300[+]HD-TV	2/3	Imagestore
v0	X							
v1	X							
W0	X							
W1	X	X				1.14.1		
W2	X	X				1.14.1		
W3	X							
W4	X							
W5	X							
W7		X				1.14.1		
W8		X				1.14.1		
WP						1.14.1		
WR						1.14.1		
k0	X	X				1.14.1		
k1	X	X				1.14.1		
k2	X	X				1.14.1		
k3		X				1.14.1		
k4		X				1.14.1		
i0	X	X				1.14.1	2.1	2.1
i1	X	X				1.14.1	2.1	2.1
i2	X	X				1.14.1	2.1	2.1
i3	X	X				1.14.1	2.1	2.1
X0	X	X				1.14.1	2.1	2.1
X1	X	X	X			1.14.1		
X2	X	X	X			1.14.1	2.1	2.1
X3	X	X	X			1.14.1	2.1	2.1
X4	X	X				1.14.1	2.1	2.1
X5	X	X				1.14.1	2.1	2.1
X6	X	X				1.14.1		
X7	X	X				1.14.1	2.1	2.1
X8	X	X				1.14.1	2.1	2.1

Command					LGK-3901	
X9	X	X			1.14.1	2.1
XA	X	X			1.14.1	2.1
XB		X			1.14.1	2.1
XC		X	X		1.14.1	2.1
XE		X			1.14.1	2.1
XH		X			1.14.1	2.1
XI		X	Xm		1.14.1	2.1
XJ		X	X		1.14.1	2.1
XK	X	X	X	4.0	1.14.1	2.1
XL		X	X		1.14.1	2.1
XM		X			1.14.1	2.1
XN		X	X		1.14.1	2.1
XP		Xi			1.14.1	
XR		X	X		1.14.1	2.1
XS		X	X		1.14.1	2.1
XT					1.14.1	
XU					1.14.1	2.1
XV					1.14.1	
XX					3.0	2.1
Xc		X			1.14.1	2.1
Xe					1.14.1	2.1
Xi						2.1
Xs		X	X		1.14.1	2.1
Xt					1.14.1	2.1
Xv					2.0.1	
Xx					3.0	2.1
j0	X	X			1.14.1	
j0 0	X	X			1.14.1	
j0 1	X	X			1.14.1	
j0 2	X	X			1.14.1	
j0 3	X	X			1.14.1	
j0 4	X	X			1.14.1	
j0 5	X	X			1.14.1	

Command	LGK-3901	DSK-3901	Imagestore 750	IntuitionXG	Intuition[+]	Imagestore 300[+]HD-TV	Imagestore 2/3
j1	X	X				1.14.1	
j1 00	X	X				1.14.1	
j1 03		X				1.14.1	
j1 10	X	X				1.14.1	
j1 2	X	X				1.14.1	
j2	X	X				1.14.1	
j2 0	X	X				1.14.1	
j2 1	X	X				1.14.1	
j2 2	X	X				1.14.1	
j3	X	X				1.14.1	
j3 0	X	X				1.14.1	
j3 1	X	X				1.14.1	
j3 2	X	X				1.14.1	
j3 3	X	X				1.14.1	
j3 4	X						
j3 5	X						
j4	X	X				1.14.1	
j40	X	X				1.14.1	
j41	X	X				1.14.1	
j42	X	X				1.14.1	
j43	X	X				1.14.1	
j44	X	X				1.14.1	
j45	X	X				1.14.1	
j46	X	X				1.14.1	
j47	X	X				1.14.1	
j5	X	X				1.14.1	
j50	X	X				1.14.1	
j51	X	X				1.14.1	
j52	X	X				1.14.1	
j53		X				1.14.1	
j6	X	X				1.14.1	
j7	X	X				1.14.1	
j70	X	X				1.14.1	
j71	X	X				1.14.1	

Command	LGK-3901	DSK-3901	Imagestore 750	IntuitionXG	Intuition[+]	Imagestore 300[+]HD-TV	Imagestore 2/3
j72	X	X			1.14.1		
j73	X	X			1.14.1		
j74	X	X			1.14.1		
j75	X	X			1.14.1		
j76					1.14.1		
j77					1.14.1		
j78		X			1.14.1		
j79					1.14.1		
j8	X						
j9	X						
ja	X	X			1.14.1		
jb	X	X			1.14.1		
jc	X	X			1.14.1		
jd	X	X			1.14.1		
je	X	X			1.14.1		
jf	X	X			1.14.1		
jg	X	X			1.14.1		
jh					1.14.1		
ji					1.14.1		
jl		X			1.141.		
o0					2.0.1		
o1					2.0.1		
o2					2.0.1		
o3					2.0.1		
o4					2.0.1		
o5					2.0.1		
o6					2.0.1		
o7					2.0.1		
o8					2.0.1		
o9					2.0.1		
oA					2.0.1		
oB					2.0.1		
oC					2.0.1		

Command	LGK-3901	DSK-3901	Imagestore 750	IntuitionXG	Intuition[+]	Imagestore 300[+]HD-TV	Imagestore 2/3
oD					2.0.1		
oE					2.0.1		
oF					2.0.1		
oG					2.0.1		
oH					2.0.1		
oI					2.0.1		
oJ					2.0.1		
oK					2.0.1		
oL					2.0.1		
oM					2.0.1		
oN					2.0.1		
oO					2.0.1		
oP					2.0.1		
oQ					2.0.1		
oR					2.0.1		
oS					2.0.1		
oT					2.0.1		
oU					2.0.1		
q0					2.0.1		
q1					2.0.1		
q2					2.0.1		
q3					2.0.1		
Y0	X	X			1.14.1		
Y1	X	X			1.14.1	2.1	2.1
Y3	X	X			1.14.1	2.1 b	2.1 b
Y6	X	X	X	4.0	1.14.1	2.1	2.1
Y7	X	X	X	4.0	1.14.1	2.1	2.1
Y8	X	X			1.14.1		
Y9	X	X	X		1.14.1		2.1
YA		X			1.14.1		2.1
YB		1.13.8	1.13.8	4.2	1.14.1		2.1
YD						2.1	2.1

Command	LGK-3901		
YO	X		
Ya	X	X	X
Yb	X		
Yc	X		
Yd	X		
Ye	X	X	X
Yf			
Yg			1.13.8
Yj			1.13.8
3 (tally)		X	
Z0	X	X	X
Z1	X	X	X
Z2	X	X	X
Z3	X	X	X
Z4	X	X	X
Z5	X	X	
Z6	X	X	X
Z7	X	X	X
Z8	X	X	X
Z9	X	X	X
ZA	X	X	X
ZB	X	X	X
ZC	X	X	X
ZD	X	X	
ZE	X	X	X
ZF	X	X	X
Za	X	X	X
Zb	X	X	X
Zc	X	X	X
Zd	X	X	X
Ze	X	X	X
Zf			X
Zg		X	X

Command	Zh	2/3	Imagestore 300[+]HD-TV	Intuition[+]	LGK-3901	DSK-3901	Imagestore 750	IntuitionXG	LGK-3901
Zh				X					2.1
n1	X		X			1.14.1			2.1
n2	X		X			1.14.1			2.1
n3	X		X			1.14.1			2.1
n4	X		X			1.14.1			
V0					4.2				
V1					4.2				
V2					4.2				
V3					4.2				
V4					4.2				
V5					4.2				
V8					4.2				
Ra	X								
Rb	X								2.1

Notes

- a: IS2-based units support key source = 'NONE'
- b: Only enquire is supported for DSK-3901 and LGK-3901
- i: Imagestore 300+ only
- k: Imagestore HD-TV only
- m: Can only be used for enquire

Audio Commands

All audio commands apply only to units fitted with Easysound option.

Appendix B

Layer Numbers

Imagestore IS1 & Imagestore IS2

Swap / Preview Mode	0 for preview layer, 1 for program layer
Cascade Mode	0 for midground layer, 1 for foreground layer

Imagestore Upstream

0 for preview layer, 1 for program layer

Imagestore 300[+] & Imagestore HD-TV

0 = Keyer 1 (DSK1)

1 = Keyer 2 (DSK2)

Intuition[+]

0 – f = Layers 1 to 16.

Imagestore 750

0 = Keyer 1 (DSK1)

1 = Keyer 2 (DSK2)

2 = Keyer 1 (DSK3)

3 = Keyer 2 (DSK4)

Each keyer can either be assigned an internal media file or an external fill/key - (similar to previous Imagestore products).

The possible configurations (when the DVE is licensed) are as follows:

DSK1 DSK2 **DVE** DSK3 DSK4

DSK1 **DVE** DSK2 DSK3 DSK4

DVE DSK1 DSK2 DSK3 DSK4

Presmaster only needs to deal with layers 0-3. It can always assume that DSK1 is the most upstream active keyer, and that DSK4 is the most downstream active keyer.

LGK-3901

0 = Keyer 1 (DSK1)

1 = Keyer 2 (DSK2)

2 = Keyer 1 (DSK3)

3 = Keyer 2 (DSK4)

4 = Keyer 2 (DSK5)

The LGK-3901 has three stores, that can be assigned to any three keyers. See the Xi command for more details.

DSK-3901

0 = Keyer 1 (DSK1)

1 = Keyer 2 (DSK2)

Appendix C

CRC Generation

The following C example demonstrates simple codes to format and send messages to an Imagestore. Imagestore status returns and acknowledges are ignored.

```
// Note these codes bear no relation to the ASCII defined
// codes with similar names.

#define STX0          0x002
#define STX1          0x003
#define ACK0          0x004
#define ACK1          0x005
#define NAK           0x007

char stx = STX0;

const UINT lstab[] =
{
    0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0,
    0x0280, 0xc241, 0xc601, 0x06c0, 0x0780, 0xc741,
    0x0500, 0xc5c1, 0xc481, 0x0440
};

const UINT mstab[] =
{
    0x0000, 0xcc01, 0xd801, 0x1400, 0xf001, 0x3c00,
    0x2800, 0xe401, 0xa001, 0x6c00, 0x7800, 0xb401,
    0x5000, 0x9c01, 0x8801, 0x4400
};

void do_crc(INT8 ch, UINT16 * crcptr)
{
    UINT tmp;

    tmp = *crcptr ^ ch;
    *crcptr = mstab[(tmp>>4) & 0xf] ^ lstab[tmp&0xf]
            ^ ((*crcptr) >> 8);
}

// Send a single command to an Imagestore, using printf style
// formatting.
void remote_send(char * format,...)
{
    UINT16 rem_crc = 0;
    INT ch;
```

```
char message[128];
char* messageptr = (char*)message;
va_list argptr;

va_start(argptr, format);
vsprintf(message, format, argptr);
va_end(argptr);

rem_send_char(stx);

while((ch = *messageptr++) != 0)
{
    rem_send_char(ch);
    do_crc(ch, &rem_crc);
}

rem_send_char(':' );
do_crc(':', &rem_crc);

rem_send_char(rem_crc & 0xff);
rem_send_char(rem_crc >> 8);

if (stx == STX0)
    stx = STX1;
else
    stx = STX0;
}
```

Appendix D

Automation Examples

Whilst it is all very well to have a list of automation commands, individual commands are often only useful when used in conjunction with other related commands. It is the job of automation to sequence automation commands at appropriate time intervals to achieve the desired on-air effects. This chapter therefore lists some basic sequences of automation commands to illustrate some basic video operations.

Please note that automation commands can also be “batched” together via GPI macros.

Keyer Animation Cut/Fade

In the following example, an animation called “Example.oxa” is loaded into DSK 2, cut up, held on-air for a period of time, faded down over 30 fields and then unloaded.

```
R01Example.oxa: // Load 'Example.oxa' into DSK 2
                  // Wait for image to load
31 1:           // Cut up DSK 2
                  // Wait with image on-air
B1 1 le:        // Set DSK 2 fade rate 60 fields
11 0:           // Fade down DSK 2
                  // Wait for fade-down to complete
A1:             // Erase DSK 2
```

AB Mixing

The following example starts by performing a cross-fade from A to B over 120 fields. It then performs an asymmetric V-fade back to A via a black colour field – again over 120 fields (40 + 80). This example assumes that the system has been set up with SDI-A and SDI-B routed into the AB mixer, and the output begins on A.

```
U603:          // Set AB transition type: X-fade
U5078:          // Fade AB over 120 fields
                // Wait
UD000000:      // Set V-fade colour to black
U8028050:      // V-fade AB over 40 fields then
                // 80 fields
```

DVE Moves

The following example brings the DVE in circuit, and then loads and plays DVE sequence number 10 forwards over its default duration. We assume that this DVE sequence begins with a full-screen keyframe so that there will be no noticeable change on-air when the DVE is loaded. After waiting for some time after the DVE sequence complete, the DVE sequence is then played in reverse direction back to the first (full-screen) keyframe. On completion of the reverse sequence, the DVE is then taken out of circuit.

```
W104:          // Put DVE in circuit
W20A3E7:      // Load first key frame of seq 10
W7001:          // Play DVE forward
                // Wait some time
W8001:          // Play DVE backward
                // Wait for reverse move complete
W100:          // Set DVE mode to None (delay)
```